

## new/exception\_lists/check\_rtime

1

```

*****
9982 Thu Feb 22 01:20:05 2018
new/exception_lists/check_rtime
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
25 #
26 #
27 # This file provides exceptions to the usual rules applied to ELF objects by
28 # check_rtime. All strings are Perl regular expressions that are compared to
29 # file paths. In addition to the standard Perl syntax, there is one extension:
30 #
31 #     MACH(dir)
32 #
33 # is expanded into a regular expression that matches the given
34 # directory, or a 64-bit subdirectory of the directory with the
35 # name of a 64-bit architecture. For example, MACH(lib) will match
36 # any of the following:
37 #
38 #     lib
39 #     lib/amd64
40 #     lib/sparcv9
41 #
42 #
43 # Directory hierarchies to skip completely
44 SKIP      ^usr/lib/libc/          # optimized libc
45 SKIP      ^usr/lib/rcm/          # 4426119
46 SKIP      ^usr/perl5/           # alan's taking care of these :-
47 SKIP      ^usr/src/             # no objects in source code
48 #
49 # Individual files that we don't examine
50 SKIP      ^boot/grub/bin/grub$
51 # USIII specific extns. cause ldd noise on USII bld. m/c
52 SKIP      ^usr/lib/fps/sun4u/UltraSPARC.*/*fptest$
53 SKIP      ^usr/MACH(lib)/lddstub$ # lddstub has no dependencies
54 SKIP      ^usr/MACH(lib)/libssagent\.\so\.\1$ # 4328854

```

## new/exception\_lists/check\_rtime

2

```

55 SKIP      ^usr/lib/MACH(iconv)/geniconvtbl.\so$ # 4384329
56 #
57 # picl file exclusions (4385799)
58 SKIP      ^usr/platform/.*/*libpsvcplugin_psr\.\so\.\1
59 SKIP      ^usr/platform/.*/*libpsvcpolicy_psr\.\so\.\1
60 SKIP      ^usr/platform/.*/*libpsvcpolicy\.\so\.\1
61 SKIP      ^usr/lib/sysevent/modules/picl_slm.\so$
62 #
63 # Objects that are allowed to have executable data segments
64 EXEC_DATA ^MACH(lib)/ld\.\so\.\1$
65 EXEC_DATA ^lib/libc\.\so\.\1$ # 6524709, 32-bit, needed for x86 only
66 EXEC_DATA ^lib/amd64/libumem\.\so\.\1$ # ptcumem
67 EXEC_DATA ^lib/libumem\.\so\.\1$ # ptcumem
68 EXEC_DATA ^opt/SUNWdtrt/tst/.*/*ustack/tst\.\helper\.\exe$
69 EXEC_DATA ^platform/.*/*MACH(kernel)/unix$
70 EXEC_DATA ^platform/.*/*multiboot$
71 #
72 # Objects that are allowed to have an executable stack
73 EXEC_STACK ^platform/.*/*MACH(kernel)/unix$
74 EXEC_STACK ^platform/.*/*multiboot$
75 EXEC_STACK ^opt/os-tests/tests/secsflags/stacky$
76 #
77 # Objects for which we allow relocations to the text segment
78 TEXTREL   ^platform/.*/*MACH(kernel)/unix$
79 #
80 # Directories and files that are allowed to have no direct bound symbols
81 NODIRECT  ^platform/.*/*MACH(kernel)/unix$
82 NODIRECT  ^usr/ucb
83 NODIRECT  ^usr/4lib/sbcp$
84 #
85 # Identify any files that should be skipped when building a crle(1)
86 # configuration file. As the hwcap libraries can be loop-back mounted onto
87 # libc, these can confuse crle(1) because of their identical dev/inode.
88 NOCRLEALT ^usr/lib/libc/libc_hwcap[1-3]\.\so\.\1$
89 #
90 # Files that should contain debugging information.
91 STAB      ^platform/.*/*MACH(kernel)/unix$
92 #
93 # Files that are allowed undefined references
94 UNDEF_REF ^usr/lib/libnisdb\.\so\.\2$
95 #
96 # Objects allowed to have unused dependencies
97 UNUSED_DEPS ^usr/lib/picl/plugins/ # require devtree dependencies
98 #
99 # libm.so.2 dependency
100 UNUSED_OBJ unused object=.*MACH(lib)/libm_hwcap1\.\so\.\2
101 #
102 # libnetsnmphelpers.so is empty in some net-snmp versions
103 UNUSED_OBJ unused object=.*/*libnetsnmphelpers\.\so\.\.*
104 UNREF_OBJ unreferenced object=.*/*libnetsnmphelpers\.\so\.\.*
105 #
106 # Unused runpaths due to dlopen() use
107 UNUSED_RPATH /usr/lib/fs/autofs.*\ from\ .\automountd
108 UNUSED_RPATH /etc/ppp/plugins.*\ from\ .\pppd
109 UNUSED_RPATH /usr/lib/inet/ppp.*\ from\ .\pppd
110 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libipsecutil\.\so\.\1
111 UNUSED_RPATH /usr/platform/.*rsmlib.*\ from\ .*librsm\.\so\.\2
112 UNUSED_RPATH $ORIGIN.*\ from\ .*fcode.so
113 UNUSED_RPATH /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgt\.\so\.\1
114 #
115 # Unused runpaths in picl code
116 UNUSED_RPATH /usr/platform/.*\ from\ .*/*usr/platform
117 UNUSED_RPATH /usr/lib/picl/.*\ from\ .*/*usr/platform
118 UNUSED_RPATH /usr/platform/.*\ from\ .*/*usr/lib/picl
119 #
120 # Unused runpaths in non-OSNET objects we can't change

```

## new/exception\_lists/check\_rtime

```

121 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnss3\.so
122 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnssutil3\.so
123 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libsmime3\.so
124 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libssl3\.so
125 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-1\.so\..3
126 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.so\..2
127 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libglib-2\.0\.so\..0
128 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libglib-2\.0\.so\..0
129 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgobject-2\.0\.so\..0
130 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgobject-2\.0\.so\..0
131 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgthread-2\.0\.so\..0
132 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgthread-2\.0\.so\..0
133 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libcrypto\.so\..0\..9\..8
134 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libnetsnp\.so\..*
135 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgcc_s\.so\..1
136 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*libgcc_s\.so\..1
137 UNUSED_RPATH /usr/lib.*\ from\ .*libgcc_s\.so\..1
138 UNUSED_RPATH /usr/postgres/8.3/lib.*\ from\ .*libpq\.so\..5
139 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libpq\.so\..5
140 UNUSED_RPATH /usr/lib.*\ from\ .*usr/lib/mps
141 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*usr/lib/mps
142 UNUSED_RPATH /usr/gnu/lib.*\ from\ .*usr/lib/libpython2\..
143 UNUSED_RPATH /usr/gnu/lib.*\ from\ .*usr/lib/64/libpython2\..
144 UNUSED_RPATH /usr/snadm/lib.*\ from\ .*usr/snadm/lib/libspmcommon\.so\..1

```

```

146 # Unused runpaths for reasons not captured above
147 UNUSED_RPATH /usr/lib/smb.*\ from\ .*libsmb\.so\..1 # future needs
149 UNUSED_RPATH /usr.*\ from\ .*tst\.gcc\.exe # gcc built

```

## 149 # Unreferenced objects of non-OSnet objects we can't change

```

150 UNREF_OBJ /lib.*\ of\ .*libcimapi\.so
151 UNREF_OBJ /lib.*\ of\ .*libdbus-1\.so\..3
152 UNREF_OBJ /lib.*\ of\ .*libdbus-glib-1\.so\..2
153 UNREF_OBJ /lib.*\ of\ .*libgio-2.0\.so\..0
154 UNREF_OBJ /lib.*\ of\ .*libglib-2.0\.so\..0
155 UNREF_OBJ /lib.*\ of\ .*libgobject-2.0\.so\..0
156 UNREF_OBJ /lib.*\ of\ .*libgthread-2.0\.so\..0
157 UNREF_OBJ /lib.*\ of\ .*libjvm\.so
158 UNREF_OBJ /lib.*\ of\ .*libnetsnp\.so\..*
159 UNREF_OBJ /lib.*\ of\ .*libnetsnpagent\.so\..*
160 UNREF_OBJ /lib.*\ of\ .*libnetsnmpmibs\.so\..*
161 UNREF_OBJ /lib.*\ of\ .*libnetsnmphelpers\.so\..*
162 UNREF_OBJ /lib.*\ of\ .*libnspr4\.so
163 UNREF_OBJ /lib.*\ of\ .*libpq\.so\..5
164 UNREF_OBJ /lib.*\ of\ .*libsoftokn3\.so
165 UNREF_OBJ /lib.*\ of\ .*libspmcommon\.so\..1
166 UNREF_OBJ /lib.*\ of\ .*libspmcommon\.so\..1
167 UNREF_OBJ /lib.*\ of\ .*libssl3\.so
168 UNREF_OBJ /lib.*\ of\ .*libtspi\.so\..1
169 UNREF_OBJ /lib.*\ of\ .*libxml2\.so\..2
170 UNREF_OBJ /lib.*\ of\ .*libxslt\.so\..1
171 UNREF_OBJ /lib.*\ of\ .*libpq\.so\..4
172 UNREF_OBJ /lib.*\ of\ .*libpython2.4\.so\..1\..0
173 UNREF_OBJ /lib.*\ of\ .*libpython2.6\.so\..1\..0
174 UNREF_OBJ /lib.*\ of\ .*libpython2.7\.so\..1\..0
175 UNREF_OBJ /libgcc_s.*\ of\ .*libstdc\+\.\.so\..6
176 UNREF_OBJ /libgcc_s.*\ of\ .*libgmodule-2\.0\.so\..0

```

## 178 # Unreferenced object of objects we can't change for other reasons

```

179 UNREF_OBJ /libmapmalloc\.so\..1;\ unused\ dependency\ of # interposer
180 UNREF_OBJ /libstdc\+\.\.so\..6;\ unused\ dependency\ of # gcc build
181 UNREF_OBJ /libgcc_s\.so\..1.*\ of\ .*libstdc\+\.\.so\..6 # omnios gcc mix
182 UNREF_OBJ /libm\.so\..2.*\ of\ .*libstdc\+\.\.so\..6 # gcc build
183 UNREF_OBJ /lib.*\ of\ .*lib/picl/plugins/ # picl

```

3

## new/exception\_lists/check\_rtime

```

184 UNREF_OBJ /lib.*\ of\ .*kcfid # interposer
185 UNREF_OBJ /libpkcs11\.so\..1; .*\ of\ .*libkmf\.so\..1 # interposed
186 # Referenced by the Studio build, not the GCC build. GCC eliminates the unused
187 # statics which have the dependence.
188 UNREF_OBJ /libc\.so\..1.*\ of\ .*kldap\.so\..1

```

```

191 # Objects that used to contain system functionality that has since
192 # migrated to libc. We preserve these libraries as pure filters for
193 # backward compatibility but nothing needs to link to them.
194 OLDDEP libaio\.so\..1 # onnv build 44
195 OLDDEP libdl\.so\..1 # on10 build 49
196 OLDDEP libdoor\.so\..1 # onnv build 12
197 OLDDEP libintl\.so\..1 # on297 build 7
198 OLDDEP libpthread\.so\..1 # on10 build 53
199 OLDDEP librt\.so\..1 # onnv build 44
200 OLDDEP libsched\.so\..1 # on10 build 36
201 OLDDEP libthread\.so\..1 # on10 build 53
202 OLDDEP libw\.so\..1 # on297 build 7

```

```

204 # Files for which we skip checking of duplicate addresses in the
205 # symbol sort sections. Such exceptions should be rare --- most code will
206 # not have duplicate addresses, since it takes assembler or a "#pragma weak"
207 # to do such aliasing in C. C++ is different: The compiler generates aliases
208 # for implementation reasons, and the mangled names used to encode argument
209 # and return value types are difficult to handle well in mapfiles.
210 # Furthermore, the Sun compiler and gcc use different and incompatible
211 # name mangling conventions. Since ON must be buildable by either, we
212 # would have to maintain two sets of mapfiles for each such object.
213 # C++ use is rare in ON, so this is not worth pursuing.
214 #

```

```

215 NOSYMSORT opt/SUNWdtrt/tst/common/pid/tst.weak2.exe # DTrace test
216 NOSYMSORT lib/amd64/libnsl\.so\..1 # C++
217 NOSYMSORT lib/sparcv9/libnsl\.so\..1 # C++
218 NOSYMSORT lib/sparcv9/libfru\.so\..1 # C++
219 NOSYMSORT usr/lib/lms # C++
220 NOSYMSORT ld\.so\..1 # libc_pic.a use
221 NOSYMSORT lib/libsun_fc\.so\..1 # C++
222 NOSYMSORT lib/amd64/libsun_fc\.so\..1 # C++
223 NOSYMSORT lib/sparcv9/libsun_fc\.so\..1 # C++
224 NOSYMSORT usr/lib/amd64/libfru\.so\..1 # C++

```

```

227 # The libprtdiag_psr.so.1 objects built under usr/src/lib/libprtdiag_psr
228 # are a family, all built using the same makefile, targeted at different
229 # sparc hardware variants. There are a small number of cases where this
230 # one size fits all approach causes an object to be linked against an
231 # unneeded library.
232 UNREF_OBJ lib/(libdevinfo|libcfgadm)\.so\..1; .*\ of\ .*SUNW,Netra-CP2300/1

```

4

new/usr/src/Makefile

1

```
*****
7333 Thu Feb 22 01:20:07 2018
new/usr/src/Makefile
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2015 OmniTI Computer Consulting, Inc. All rights reserved.
27 # Copyright 2016 Toomas Soome <tsoome@me.com>
28 # Copyright 2017 Joyent, Inc.
29 #
30 #
31 #
32 # Makefile for system source
33 #
34 # include global definitions
35 include Makefile.master
36 #
37 # the Targetdirs file is the AT&T target.dirs file in a makefile format.
38 # it defines TARGETDIRS and ROOTDIRS.
39 include Targetdirs
40 #
41 COMMON_SUBDIRS= data uts lib cmd ucplib ucblib psm man test
42 sparc_SUBDIRS= stand
43 i386_SUBDIRS= grub boot
44 #
45 #
46 # sparc needs to build stand before psm
47 #
48 $(SPARC_BLD)psm: stand
49 #
50 SUBDIRS= $(COMMON_SUBDIRS) $($(MACH)_SUBDIRS)
51 #
52 HDRSUBDIRS= uts head lib cmd
53 #
54 # UCB headers are bug-for-bug compatible and not checkable against the header
```

new/usr/src/Makefile

2

```
55 # standards.
56 #
57 CHKHDRSUBDIRS= head uts lib
58 #
59 #
60 # Headers that can be built in parallel
61 #
62 PARALLEL_HEADERS = sysheaders userheaders libheaders cmdheaders
63 #
64 #
65 # Directories that can be built in parallel
66 #
67 PARALLEL_DIRS = data uts lib man
68 #
69 # The check target also causes smf(5) service manifests to be validated.
70 CHKMFSTSUBDIRS= cmd
71 #
72 # And man page formats
73 CHKMANSUBDIRS = man
74 #
75 MSGSUBDIRS= cmd ucblib lib data
76 DOMAINS= \
77     SUNW_OST_ADMIN \
78     SUNW_OST_NETRPC \
79     SUNW_OST_OSCMD \
80     SUNW_OST_OSLIB \
81     SUNW_OST_UCBCMD \
82     SUNW_OST_ZONEINFO
83 #
84 MSGDDIRS= $(DOMAINS:%=$(MSGROOT)/%)
85 MSGDIRS= $(MSGROOT) $(MSGDDIRS) $(MSGROOT)/LC_TIME
86 #
87 all := TARGET= all
88 install := TARGET= install
89 install1 := TARGET= install
90 install2 := TARGET= install
91 install_h := TARGET= install_h
92 clean := TARGET= clean
93 clobber := TARGET= clobber
94 check := TARGET= check
95 #
96 .KEEP_STATE:
97 #
98 #
99 # Note: install does not cause a build in pkg. To build packages,
100 # cd pkg and do a 'make install'
101 #
102 #
103 all: mapfiles closedbins sgs .WAIT $(SUBDIRS) pkg
104 #
105 #
106 # The _msg build is a two-step process. First, the _msg dependency
107 # causes recursive makes in $(MSGSUBDIRS), which stages raw message
108 # files in $(ROOT)/catalog. Second, the action from the install
109 # target rule causes those messages to be post-processed from where
110 # they were staged in $(ROOT)/catalog, and the results placed into the
111 # proto area.
112 #
113 # The stage-licenses target causes the license files needed for
114 # packaging to be pulled from $(SRC) and $(CLOSED) and staged in
115 # $(ROOT)/licenses.
116 #
117 install: install1 install2 _msg stage-licenses
118 @cd msg; pwd; $(MAKE) _msg
119 @rm -rf "$(ROOT)/catalog"
```

## new/usr/src/Makefile

3

```

121 stage-licenses: install2
122     @cd pkg; pwd; $(MAKE) stage-licenses

124 install1: mapfiles closedbins sgs

126 install2: install1 .WAIT $(SUBDIRS)

128 _msg: _msgdirs rootdirs FRC
129     @for m in $(MSGSUBDIRS); do \
130         cd $$m; pwd; $(MAKE) _msg; cd ..; \
131     done

133 mapfiles: bldtools
134     @cd common/mapfiles; pwd; $(MAKE) install

136 clean: $(SUBDIRS) head pkg
137 clobber: $(SUBDIRS) head pkg clobber_local
138 clobber_local:
139     @cd tools; pwd; $(MAKE) clobber
140     @cd common/mapfiles; pwd; $(MAKE) clobber
141     @cd msg; pwd; $(MAKE) clobber

143 # If the tarballs are included inside ON_CLOSED_BINS, use those to extract and
144 # preserve the permissions (in case a distro ships them).

146 closedbins: bldtools $(ROOTDIRS) FRC
147     @CLOSED_ROOT="$${ON_CLOSED_BINS}/root_$(MACH)$$${RELEASE_BUILD+nd}"; \
148     if [ -f "$${ON_CLOSED_BINS}/on-closed-bins$$${RELEASE_BUILD+nd}.$(MACH).tar" ]; then \
149         $(ECHO) "Extracting tarball $${ON_CLOSED_BINS}/on-closed-bins$$${RELEASE_BUILD+nd}.$(MACH).tar"; \
150         (cd $(CODEMGR_WS); \
151             $(TAR) xjpf $${ON_CLOSED_BINS}/on-closed-bins$$${RELEASE_BUILD+nd}.$(MACH).tar; \
152             CLOSED_ROOT="$${CODEMGR_WS}/closed/root_$(MACH)$$${RELEASE_BUILD+nd}"; \
153         fi; \
154         if [ ! -d "$${CLOSED_ROOT}" ]; then \
155             $(ECHO) "Error: ON_CLOSED_BINS must point to closed" \
156                 "binaries."; \
157             $(ECHO) "root_$(MACH)$$${RELEASE_BUILD+nd} is not" \
158                 "present in $${ON_CLOSED_BINS}."; \
159             exit 1; \
160         fi; \
161         $(ECHO) "Copying closed binaries from $${CLOSED_ROOT}"; \
162         (cd $${CLOSED_ROOT}; \
163             $(TAR) cfx - $(CODEMGR_WS)/exception_lists/closed-bins .) | \
164             (cd $(ROOT); $(TAR) xBpf -); \
165         ( cd $(ROOT); $(CTFSTRIP) $$(cd $${CLOSED_ROOT}; $(FIND) \
166             ./kernel ./usr/kernel ./platform/*/kernel -type f -a -perm -u+x | \
167             $(EGREP) -vf $(CODEMGR_WS)/exception_lists/closed-bins) )

169 #
170 # Declare what parts can be built in parallel
171 # DUMMY at the end is used in case macro expansion produces an empty string to
172 # prevent everything going in parallel
173 #
174 .PARALLEL: $(PARALLEL_HEADERS) DUMMY
175 .PARALLEL: $(PARALLEL_DIRS) DUMMY

177 $(SUBDIRS) head pkg: FRC
178     @cd $@; pwd; $(MAKE) $(TARGET)

180 # librpcsvc has a dependency on headers installed by
181 # userheaders, hence the .WAIT before libheaders.
182 sgs: rootdirs .WAIT sysheaders userheaders .WAIT \
183     libheaders cmdheaders

185 #
186 # Top-level setup target to setup the development environment that includes

```

## new/usr/src/Makefile

4

```

187 # headers, tools and generated mapfiles. For open-only builds (i.e.: source
188 # trees w/o usr/closed), this also depends on the closedbins target (above)
189 # in order to properly seed the proto area. Note, although the tools are
190 # dependent on a number of constant mapfiles, the tools themselves are
191 # required to build the generated mapfiles.
192 #
193 setup: closedbins bldtools sgs mapfiles

195 bldtools:
196     @cd tools; pwd; $(MAKE) install

198 # /var/mail/:saved is a special case because of the colon in the name.
199 #
200 rootdirs: $(ROOTDIRS)
201     $(INS) -d -m 775 $(ROOT)/var/mail/:saved

203 lint: FRC
204     $(MAKE) -f Makefile.lint

206 _msgdirs:     $(MSGDIRS)

208 $(ROOTDIRS) $(MSGDIRS):
209     $(INS.dir)

211 userheaders: FRC
212     @cd head; pwd; $(MAKE) install_h

214 libheaders: bldtools
215     @cd lib; pwd; $(MAKE) install_h

217 sysheaders: FRC
218     @cd uts; pwd; $(MAKE) install_h

220 cmdheaders: FRC
221     @cd cmd/devfsadm; pwd; $(MAKE) install_h
222     @cd cmd/fm; pwd; $(MAKE) install_h
223     @cd cmd/mdb; pwd; $(MAKE) install_h

225 check: $(CHKHDRSUBDIRS) $(CHKMFSTSUBDIRS) $(CHKMANSUBDIRS)

227 #
228 # Cross-reference customization: skip all of the subdirectories that
229 # don't contain actual source code.
230 #
231 XRPRUNE = pkg prototypes
232 XRINCDIRS = uts/common head uchead

234 cscope.out tags: FRC
235     $(XREF) -f -x $@

237 FRC:

239 #
240 # Targets for reporting compiler versions; nightly uses these.
241 #

243 cc-version:
244     @$(ECHO) 32-bit compiler; \
245     @if $(MACH_CC) -_versions >/dev/null 2>/dev/null; then \
246         $(ECHO) 32-bit compiler; \
247         $(ECHO) $$(MACH_CC); \
248         $(CW) --versions $$(CW_CC_COMPILERS) 2>&1 | \
249         $(EGREP) '^cw/cc/gcc/primary/shadow'; \
250     else \
251         _COMPILER='$$(MACH_CC) -_compiler 2>/dev/null || $(TRUE)'; \

```

```

251     if [ -z "$$_COMPILER" ]; then           \
252         $(ECHO) No 32-bit compiler found;   \
253         exit 1;                             \
254     else                                     \
255         $(ECHO) 32-bit compiler;           \
256         $(ECHO) ${$(MACH)_CC};             \
257         $(ECHO) $$_COMPILER;              \
258         ${$(MACH)_CC} -V 2>&1 | head -1;    \
259     fi;                                     \
260     fi;

```

```

248 cc64-version:
249     @$(ECHO) 64-bit compiler;               \
250     $(ECHO) ${$(MACH)_CC};                 \
251     $(CW) --versions ${CW_CC_COMPILERS} 2>&1
263     @if ${$(MACH64)_CC} -_versions >/dev/null 2>/dev/null; then \
264         $(ECHO) 64-bit compiler;           \
265         $(ECHO) ${$(MACH64)_CC};           \
266         ${$(MACH64)_CC} -_versions 2>&1 |  \
267         $(EGREP) '^$(cw|cc/gcc/primary|shadow)'; \
268     else                                     \
269         __COMPILER='${$(MACH64)_CC} -_compiler 2>/dev/null || $(TRUE)'; \
270         if [ -z "$$_COMPILER" ]; then      \
271             $(ECHO) No 64-bit compiler found; \
272             exit 1;                         \
273         else                                 \
274             $(ECHO) 64-bit compiler;       \
275             $(ECHO) ${$(MACH64)_CC};       \
276             $(ECHO) $$_COMPILER;          \
277             ${$(MACH64)_CC} -V 2>&1 | head -1; \
278         fi;                                 \
279     fi;

```

```

253 java-version:
254     @if [ -x "$(JAVAC)" ]; then             \
255         $(ECHO) $(JAVAC);                  \
256         $(JAVA_ROOT)/bin/java -fullversion 2>&1 | head -1; \
257     else                                     \
258         $(ECHO) No Java compiler found;    \
259         exit 1;                             \
260     fi;

```

```

*****
35818 Thu Feb 22 01:20:07 2018
new/usr/src/Makefile.master
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
27 # Copyright 2015 Gary Mills
28 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
29 # Copyright 2016 Toomas Soome <tsoome@me.com>
30 #
31 #
32 #
33 # Makefile.master, global definitions for system source
34 #
35 ROOT=          /proto
36 #
37 #
38 # Adjunct root, containing an additional proto area to be used for headers
39 # and libraries.
40 #
41 ADJUNCT_PROTO=
42 #
43 #
44 # Adjunct for building things that run on the build machine.
45 #
46 NATIVE_ADJUNCT= /usr
47 #
48 #
49 # RELEASE_BUILD should be cleared for final release builds.
50 # NOT_RELEASE_BUILD is exactly what the name implies.
51 #
52 # __GNUC toggles the building of ON components using gcc and related tools.
53 # Normally set to '#', set it to '' to do gcc build.
54 #

```

```

55 # The declaration POUND_SIGN is always '#'. This is needed to get around the
56 # make feature that '#' is always a comment delimiter, even when escaped or
57 # quoted. We use this macro expansion method to get POUND_SIGN rather than
58 # always breaking out a shell because the general case can cause a noticeable
59 # slowdown in build times when so many Makefiles include Makefile.master.
60 #
61 # While the majority of users are expected to override the setting below
62 # with an env file (via nightly or bldenv), if you aren't building that way
63 # (ie, you're using "ws" or some other bootstrapping method) then you need
64 # this definition in order to avoid the subshell invocation mentioned above.
65 #
66 PRE_POUND=
67 #
68 POUND_SIGN=      pre\#
69 #
70 #
71 #
72 #
73 PATCH_BUILD=
74 #
75 #
76 #
77 #
78 #
79 #
80 INTEL_BLD=
81 #
82 # The variables below control the compilers used during the build.
83 # There are a number of permutations.
84 #
85 # __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
86 # one is not POUND_SIGN is the primary, with the other as the shadow.  They
87 # may also be used to control entirely compiler-specific Makefile assignments.
88 # __GNUC and GCC are the default.
89 #
90 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
91 # There is no Sun C analogue.
92 #
93 # The following version-specific options are operative regardless of which
94 # compiler is primary, and control the versions of the given compilers to be
95 # used.  They also allow compiler-version specific Makefile fragments.
96 #
97 #
98 #
99 #
100 #
101 #
102 # Allow build-time "configuration" to enable or disable some things.
103 # The default is POUND_SIGN, meaning "not enabled".  If the environment
104 # passes in an override like ENABLE_SMB_PRINTING= (empty) that will
105 # uncomment things in the lower Makefiles to enable the feature.
106 #
107 #
108 # CLOSED is the root of the tree that contains source which isn't released
109 # as open source
110 #
111 #
112 # BUILD_TOOLS is the root of all tools including compilers.
113 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.
114 #
115 #
116 #
117 #
118 # define runtime JAVA_HOME, primarily for cmd/pools/poold
119 #
120 # define buildtime JAVA_ROOT

```

## new/usr/src/Makefile.master

```

121 JAVA_ROOT=      /usr/java
122 # define to build with JAVA 8
123 BLD_JAVA_8=     $(POUND_SIGN)

125 GCC_ROOT=       /opt/gcc/4.4.4
126 GCCLIBDIR=      $(GCC_ROOT)/lib
127 GCCLIBDIR64=    $(GCC_ROOT)/lib/$(MACH64)

129 DOCBOOK_XSL_ROOT= /usr/share/sgml/docbook/xsl-stylesheets

131 RPCGEN=         /usr/bin/rpcgen
132 STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
133 ELFXTRACT=      $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
134 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
135 BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
136 VTFONTCVT=     $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcv
137 # echo(1) and true(1) are specified without absolute paths, so that the shell
138 # spawned by make(1) may use the built-in versions. This is minimally
139 # problematic, as the shell spawned by make(1) is known and under control, the
140 # only risk being if the shell falls back to $PATH.
141 #
142 # We specifically want an echo(1) that does interpolation of escape sequences,
143 # which ksh93, /bin/sh, and bash will all provide.
144 ECHO=           echo
145 TRUE=           true
146 INS=            $(ONBLD_TOOLS)/bin/$(MACH)/install
147 SYMLINK=        /usr/bin/ln -s
148 LN=            /usr/bin/ln
149 MKDIR=          /usr/bin/mkdir
150 CHMOD=          /usr/bin/chmod
151 MV=            /usr/bin/mv -f
152 RM=            /usr/bin/rm -f
153 CUT=           /usr/bin/cut
154 NM=            /usr/ccs/bin/nm
155 DIFF=          /usr/bin/diff
156 GREP=          /usr/bin/grep
157 EGREP=         /usr/bin/egrep
158 ELFWRAP=       /usr/bin/elfwrap
159 KSH93=         /usr/bin/ksh93
160 SED=           /usr/bin/sed
161 AWK=           /usr/bin/nawk
162 CP=            /usr/bin/cp -f
163 MCS=           /usr/ccs/bin/mcs
164 CAT=           /usr/bin/cat
165 ELFDUMP=       /usr/ccs/bin/elfdump
166 M4=            /usr/bin/m4
167 STRIP=         /usr/ccs/bin/strip
168 LEX=           /usr/ccs/bin/lex
169 FLEX=          /usr/bin/flex
170 YACC=          /usr/ccs/bin/yacc
171 CPP=           /usr/lib/cpp
172 JAVAC=         $(JAVA_ROOT)/bin/javac
173 JAVAH=         $(JAVA_ROOT)/bin/javah
174 JAVADOC=       $(JAVA_ROOT)/bin/javadoc
175 RMIC=          $(JAVA_ROOT)/bin/rmic
176 JAR=           $(JAVA_ROOT)/bin/jar
177 CTFCONVERT=    $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
178 CTFMERGE=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
179 CTFSTABS=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
180 CTFSTRIP=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
181 NDRGEN=        $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
182 GENOFFSETS=   $(ONBLD_TOOLS)/bin/genoffsets
183 XREF=          $(ONBLD_TOOLS)/bin/xref
184 FIND=          /usr/bin/find
185 PERL=          /usr/bin/perl
186 PERL_VERSION=  5.10.0

```

3

## new/usr/src/Makefile.master

```

187 PERL_PKGVERS=   -510
188 PERL_ARCH =     i86pc-solaris-64int
189 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
190 PYTHON_VERSION= 2.7
191 PYTHON_PKGVERS= -27
192 PYTHON=         /usr/bin/python$(PYTHON_VERSION)
193 SORT=           /usr/bin/sort
194 TOUCH=          /usr/bin/touch
195 WC=             /usr/bin/wc
196 XARGS=          /usr/bin/xargs
197 ELFEDIT=        /usr/bin/elfedit
198 DTRACE=         /usr/sbin/dtrace -xnolib
199 UNIQ=           /usr/bin/uniq
200 TAR=            /usr/bin/tar
201 ASTBINDIR=      /usr/ast/bin
202 MSGCC=          $(ASTBINDIR)/msgcc
203 MSGFMT=         /usr/bin/msgfmt -s
204 LCDEF=          $(ONBLD_TOOLS)/bin/$(MACH)/localedef
205 TIC=            $(ONBLD_TOOLS)/bin/$(MACH)/tic
206 ZIC=            $(ONBLD_TOOLS)/bin/$(MACH)/zic

208 FILEMODE=       644
209 DIRMODE=        755

211 # Declare that nothing should be built in parallel.
212 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
213 .NO_PARALLEL:

215 # For stylistic checks
216 #
217 # Note that the X and C checks are not used at this time and may need
218 # modification when they are actually used.
219 #
220 CSTYLE=          $(ONBLD_TOOLS)/bin/cstyle
221 CSTYLE_TAIL=    $(ONBLD_TOOLS)/bin/hdrchk
222 HDRCHK=         $(ONBLD_TOOLS)/bin/hdrchk
223 HDRCHK_TAIL=    $(ONBLD_TOOLS)/bin/jstyle
224 JSTYLE=         $(ONBLD_TOOLS)/bin/jstyle

226 DOT_H_CHECK=    \
227     @$(ECHO) "checking $<;" ; $(CSTYLE) $< $(CSTYLE_TAIL); \
228     $(HDRCHK) $< $(HDRCHK_TAIL)

230 DOT_X_CHECK=    \
231     @$(ECHO) "checking $<;" ; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
232     $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

234 DOT_C_CHECK=    \
235     @$(ECHO) "checking $<;" ; $(CSTYLE) $< $(CSTYLE_TAIL)

237 MANIFEST_CHECK= \
238     @$(ECHO) "checking $<;" ; \
239     SVCCFG_DTD=$(SRC)/cmd/svc/dtd/service_bundle.dtd.1 \
240     SVCCFG_REPOSITORY=$(SRC)/cmd/svc/seed/global.db \
241     SVCCFG_CONFIGD_PATH=$(SRC)/cmd/svc/configd/svc.configd-native \
242     $(SRC)/cmd/svc/svccfg/svccfg-native validate $<

244 INS.file=       $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
245 INS.dir=        $(INS) -s -d -m $(DIRMODE) $@
246 # installs and renames at once
247 #
248 INS.rename=     $(INS.file); $(MV) $(@D)/$(<F) $@

250 # install a link
251 INSLINKTARGET=  $<
252 INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@

```

4

```

253 INS.symlink= $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@
255 #
256 # Python bakes the mtime of the .py file into the compiled .pyc and
257 # rebuilds if the baked-in mtime != the mtime of the source file
258 # (rather than only if it's less than), thus when installing python
259 # files we must make certain to not adjust the mtime of the source
260 # (.py) file.
261 #
262 INS.pyfile= $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $

264 # MACH must be set in the shell environment per uname -p on the build host
265 # More specific architecture variables should be set in lower makefiles.
266 #
267 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
268 # architectures on which we do not build 64-bit versions.
269 # (There are no such architectures at the moment.)
270 #
271 # Set BUILD64=# in the environment to disable 64-bit amd64
272 # builds on i386 machines.

274 MACH64_1= $(MACH:sparc=sparcv9)
275 MACH64= $(MACH64_1:i386=amd64)

277 MACH32_1= $(MACH:sparc=sparcv7)
278 MACH32= $(MACH32_1:i386=i86)

280 sparc_BUILD64=
281 i386_BUILD64=
282 BUILD64= $($(_MACH)_BUILD64)

284 #
285 # C compiler mode. Future compilers may change the default on us,
286 # so force extended ANSI mode globally. Lower level makefiles can
287 # override this by setting CCMODE.
288 #
289 CCMODE= -Xa
290 CCMODE64= -Xa

292 #
293 # C compiler verbose mode. This is so we can enable it globally,
294 # but turn it off in the lower level makefiles of things we cannot
295 # (or aren't going to) fix.
296 #
297 CCVERBOSE= -v

299 # set this to the secret flag "-Wc,-Qiselect-v9abiwarn=1" to get warnings
300 # from the compiler about places the -xarch=v9 may differ from -xarch=v9c.
301 V9ABIWARN=

303 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
304 # symbols (used to detect conflicts between objects that use global registers)
305 # we disable this now for safety, and because genunix doesn't link with
306 # this feature (the v9 default) enabled.
307 #
308 # REGSYM is separate since the C++ driver syntax is different.
309 CCREGSYM= -Wc,-Qiselect-regsym=0
310 CCCREGSYM= -Qoption cg -Qiselect-regsym=0

312 # Prevent the removal of static symbols by the SPARC code generator (cg).
313 # The x86 code generator (ube) does not remove such symbols and as such
314 # using this workaround is not applicable for x86.
315 #
316 CCSTATICSYM= -Wc,-Qassembler-ounrefsym=0
317 #
318 # generate 32-bit addresses in the v9 kernel. Saves memory.

```

```

319 CCABS32= -Wc,-xcode=abs32
320 #
321 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
322 # system calls.
323 CC32BITCALLERS= -_gcc=-massume-32bit-callers

325 # GCC, especially, is increasingly beginning to auto-inline functions and
326 # sadly does so separately not under the general -fno-inline-functions
327 # Additionally, we wish to prevent optimisations which cause GCC to clone
328 # functions -- in particular, these may cause unhelpful symbols to be
329 # emitted instead of function names
330 CCNOAUTOINLINE= _gcc=-fno-inline-small-functions \
331 _gcc=-fno-inline-functions-called-once \
332 _gcc=-fno-ipa-cp

334 # One optimization the compiler might perform is to turn this:
335 # #pragma weak foo
336 # extern int foo;
337 # if (&foo)
338 #     foo = 5;
339 # into
340 #     foo = 5;
341 # Since we do some of this (foo might be referenced in common kernel code
342 # but provided only for some cpu modules or platforms), we disable this
343 # optimization.
344 #
345 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
346 i386_CCUNBOUND =
347 CCUNBOUND = $($(_MACH)_CCUNBOUND)

349 #
350 # compiler '-xarch' flag. This is here to centralize it and make it
351 # overridable for testing.
352 sparc_XARCH= -m32
353 sparcv9_XARCH= -m64
354 i386_XARCH= -m32
355 amd64_XARCH= -m64 -Ui386 -U__i386

357 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
358 sparc_AS_XARCH= -xarch=v8plus
359 sparcv9_AS_XARCH= -xarch=v9
360 i386_AS_XARCH=
361 amd64_AS_XARCH= -xarch=amd64 -P -Ui386 -U__i386

363 #
364 # These flags define what we need to be 'standalone' i.e. -not- part
365 # of the rather more cosy userland environment. This basically means
366 # the kernel.
367 #
368 # XX64 future versions of gcc will make -mcmmodel=kernel imply -mno-red-zone
369 #
370 sparc_STAND_FLAGS= -_gcc=-ffreestanding
371 sparcv9_STAND_FLAGS= -_gcc=-ffreestanding
372 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
373 # additions to SSE (SSE2, AVX, etc.)
374 NO_SIMD= -_gcc=-mno-mmx -_gcc=-mno-sse
375 i386_STAND_FLAGS= -_gcc=-ffreestanding $(NO_SIMD)
376 amd64_STAND_FLAGS= -xmodel=kernel $(NO_SIMD)

378 SAVEARGS= -Wu,-save_args
379 amd64_STAND_FLAGS += $(SAVEARGS)

381 STAND_FLAGS_32 = $($(_MACH)_STAND_FLAGS)
382 STAND_FLAGS_64 = $($(_MACH64)_STAND_FLAGS)

384 #

```



```

385 # disable the incremental linker
386 ILDOFF= -xildoff
387 #
388 XDEPEND= -xdepend
389 XFFLAG= -xF=%all
390 XESS= -xs
391 XSTRCONST= -xstrconst

393 #
394 # turn warnings into errors (C)
395 CERRWARN = -errtags=yes -errwarn=%all
396 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
397 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

399 CERRWARN += _gcc=-Wno-missing-braces
400 CERRWARN += _gcc=-Wno-sign-compare
401 CERRWARN += _gcc=-Wno-unknown-pragmas
402 CERRWARN += _gcc=-Wno-unused-parameter
403 CERRWARN += _gcc=-Wno-missing-field-initializers

405 # Unfortunately, this option can misfire very easily and unfixably.
406 CERRWARN += _gcc=-Wno-array-bounds

408 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
409 # -nd builds
410 $(RELEASE_BUILD)CERRWARN += _gcc=-Wno-unused
411 $(RELEASE_BUILD)CERRWARN += _gcc=-Wno-empty-body

413 #
414 # turn warnings into errors (C++)
415 CCERRWARN= -xwe

417 # C standard. Keep Studio flags until we get rid of lint.
418 CSTD_GNU89= -xc99=%none
419 CSTD_GNU99= -xc99=%all
420 CSTD= $(CSTD_GNU89)
421 C99LMODE= $(CSTD:-xc99%=-Xc99%)

423 # In most places, assignments to these macros should be appended with +=
424 # (CPPFLAGS.first allows values to be prepended to CPPFLAGS).
425 sparc_CFLAGS= $(sparc_XARCH) $(CSTATICSYM)
426 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
427 $(CSTATICSYM)
428 i386_CFLAGS= $(i386_XARCH)
429 amd64_CFLAGS= $(amd64_XARCH)

431 sparc_ASFLAGS= $(sparc_AS_XARCH)
432 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
433 i386_ASFLAGS= $(i386_AS_XARCH)
434 amd64_ASFLAGS= $(amd64_AS_XARCH)

436 #
437 sparc_COPTFLAG= -xO3
438 sparcv9_COPTFLAG= -xO3
439 i386_COPTFLAG= -O
440 amd64_COPTFLAG= -xO3

442 COPTFLAG= $($(_MACH)_COPTFLAG)
443 COPTFLAG64= $($(_MACH64)_COPTFLAG)

445 # When -g is used, the compiler globalizes static objects
446 # (gives them a unique prefix). Disable that.
447 CNOGLOBAL= -W0,-noglobal

449 # Direct the Sun Studio compiler to use a static globalization prefix based on t
450 # name of the module rather than something unique. Otherwise, objects

```

```

451 # will not build deterministically, as subsequent compilations of identical
452 # source will yeild objects that always look different.
453 #
454 # In the same spirit, this will also remove the date from the N_OPT stab.
455 CGLOBALSTATIC= -W0,-xglobalstatic

457 # Sometimes we want all symbols and types in debugging information even
458 # if they aren't used.
459 CALLSYMS= -W0,-xdbggen=no%usedonly

461 #
462 # Default debug format for Sun Studio 11 is dwarf, so force it to
463 # generate stabs.
464 #
465 DEBUGFORMAT= -xdebugformat=stabs

467 #
468 # Flags used to build in debug mode for ctf generation. Bugs in the Devpro
469 # compilers currently prevent us from building with cc-emitted DWARF.
470 #
471 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)
472 CTF_FLAGS_i386 = -g $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)

474 CTF_FLAGS_sparcv9 = $(CTF_FLAGS_sparc)
475 CTF_FLAGS_amd64 = $(CTF_FLAGS_i386)

477 # Sun Studio produces broken userland code when saving arguments.
478 $(__GNUCC)CTF_FLAGS_amd64 += $(SAVEARGS)

480 CTF_FLAGS_32 = $(CTF_FLAGS_$(_MACH)) $(DEBUGFORMAT)
481 CTF_FLAGS_64 = $(CTF_FLAGS_$(_MACH64)) $(DEBUGFORMAT)
482 CTF_FLAGS = $(CTF_FLAGS_32)

484 #
485 # Flags used with genoffsets
486 #
487 GOFLAGS = $(CALLSYMS) $(CDWARFSTR)
488 GOFLAGS = _noecho \
489 $(CALLSYMS) \
490 $(CDWARFSTR)

492 OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
493 $(CW) --noecho $(CW_CC_COMPILERS) -- $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)
494 $(CC) $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)

495 #
496 # tradeoff time for space (smaller is better)
497 #
498 sparc_SPACEFLAG = -xspace -W0,-Lt
499 sparcv9_SPACEFLAG = -xspace -W0,-Lt
500 i386_SPACEFLAG = -xspace
501 amd64_SPACEFLAG =

503 SPACEFLAG = $($(_MACH)_SPACEFLAG)
504 SPACEFLAG64 = $($(_MACH64)_SPACEFLAG)

506 #
507 # The Sun Studio 11 compiler has changed the behaviour of integer
508 # wrap arounds and so a flag is needed to use the legacy behaviour
509 # (without this flag panics/hangs could be exposed within the source).
510 #
511 sparc_IROPTFLAG = -W2,-xwrap_int

```

```

512 sparcv9_IROPTFLAG      = -W2,-xwrap_int
513 i386_IROPTFLAG         =
514 amd64_IROPTFLAG        =

516 IROPTFLAG              = $($MACH)_IROPTFLAG
517 IROPTFLAG64            = $($MACH64)_IROPTFLAG

519 sparc_XREGSFLAG        = -xregs=no%appl
520 sparcv9_XREGSFLAG      = -xregs=no%appl
521 i386_XREGSFLAG         =
522 amd64_XREGSFLAG        =

524 XREGSFLAG              = $($MACH)_XREGSFLAG
525 XREGSFLAG64            = $($MACH64)_XREGSFLAG

527 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
528 # avoids stripping it.
529 SOURCEDEBUG            = $(POUND_SIGN)
530 SRCDBGBLD              = $(SOURCEDEBUG:yes=)

532 #
533 # These variables are intended ONLY for use by developers to safely pass extra
534 # flags to the compilers without unintentionally overriding Makefile-set
535 # flags. They should NEVER be set to any value in a Makefile.
536 #
537 # They come last in the associated FLAGS variable such that they can
538 # explicitly override things if necessary, there are gaps in this, but it's
539 # the best we can manage.
540 #
541 CUSERFLAGS              =
542 CUSERFLAGS64            = $(CUSERFLAGS)
543 CCUSERFLAGS             =
544 CCUSERFLAGS64           = $(CCUSERFLAGS)

546 CSOURCEDEBUGFLAGS      =
547 CCSOURCEDEBUGFLAGS     =
548 $(SRCDBGBLD)CSOURCEDEBUGFLAGS = -g -xs
549 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS = -g -xs

551 CFLAGS=                 $(COPTFLAG) $($MACH)_CFLAGS $(SPACEFLAG) $(CCMODE) \
552                         $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG) \
553                         $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
554                         $(CUSERFLAGS)
555 CFLAGS64=               $(COPTFLAG64) $($MACH64)_CFLAGS $(SPACEFLAG64) $(CCMODE64) \
556                         $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG64) \
557                         $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
558                         $(CUSERFLAGS64)
559 #
560 # Flags that are used to build parts of the code that are subsequently
561 # run on the build machine (also known as the NATIVE_BUILD).
562 #
563 NATIVE_CFLAGS=          $(COPTFLAG) $($NATIVE_MACH)_CFLAGS $(CCMODE) \
564                         $(ILDOFF) $(CERRWARN) $(CSTD) $($NATIVE_MACH)_CCUNBOUND) \
565                         $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
566                         $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

568 DTEXTDOM=-DTEXT_DOMAIN="\$(TEXT_DOMAIN)"      # For messaging.
569 DTS_ERRNO=-D_TS_ERRNO
570 CPPFLAGS.first= # Please keep empty. Only lower makefiles should set this.
571 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
572                 $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
573                 $(ADJUNCT_PROTO:=-I%/usr/include)
574 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
575                 $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
576 CPPFLAGS=             $(CPPFLAGS.first) $(CPPFLAGS.master)
577 AS_CPPFLAGS=          $(CPPFLAGS.first) $(CPPFLAGS.master)

```

```

578 JAVAFLAGS=            -source 1.6 -target 1.6 -Xlint:deprecation,-options

580 #
581 # For source message catalogue
582 #
583 .SUFFIXES: $(SUFFIXES) .i .po
584 MSGROOT= $(ROOT)/catalog
585 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
586 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
587 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
588 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

590 CLOBBERFILES += $(POFILE) $(POFILES)
591 COMPILER.cpp= $(CC) -E -C $(CFLAGS) $(CPPFLAGS)
592 XGETTEXT= /usr/bin/xgettext
593 XGETTEXTFLAGS= -c TRANSLATION_NOTE
594 GNUXGETTEXT= /usr/gnu/bin/xgettext
595 GNUXGETTEXTFLAGS= --add-comments=TRANSLATION_NOTE --keyword=_ \
596                  --strict --no-location --omit-header
597 BUILD.po= $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $<.i ;\
598           $(RM) $@ ;\
599           $(SED) "/^domain/d" < $(<F).po > $@ ;\
600           $(RM) $(<F).po $<.i

602 #
603 # This is overwritten by local Makefile when PROG is a list.
604 #
605 POFILE= $(PROG).po

607 sparc_CCFLAGS=         -cg92 -compat=4 \
608                         -Qoption ccfe -messages=no%anachronism \
609                         $(CCERRWARN)
610 sparcv9_CCFLAGS=       $(sparcv9_XARCH) -dalign -compat=5 \
611                         -Qoption ccfe -messages=no%anachronism \
612                         -Qoption ccfe -features=no%conststrings \
613                         $(CCCREGSYM) \
614                         $(CCERRWARN)
615 i386_CCFLAGS=          -compat=4 \
616                         -Qoption ccfe -messages=no%anachronism \
617                         -Qoption ccfe -features=no%conststrings \
618                         $(CCERRWARN)
619 amd64_CCFLAGS=         $(amd64_XARCH) -compat=5 \
620                         -Qoption ccfe -messages=no%anachronism \
621                         -Qoption ccfe -features=no%conststrings \
622                         $(CCERRWARN)

624 sparc_CCOPTFLAG=      -O
625 sparcv9_CCOPTFLAG=    -O
626 i386_CCOPTFLAG=       -O
627 amd64_CCOPTFLAG=      -O

629 CCOPTFLAG=             $($MACH)_CCOPTFLAG
630 CCOPTFLAG64=          $($MACH64)_CCOPTFLAG
631 CCFLAGS=               $(CCOPTFLAG) $($MACH)_CCFLAGS $(CCSOURCEDEBUGFLAGS) \
632                         $(CUSERFLAGS)
633 CCFLAGS64=             $(CCOPTFLAG64) $($MACH64)_CCFLAGS $(CCSOURCEDEBUGFLAGS) \
634                         $(CCUSERFLAGS64)

636 #
637 #
638 #
639 ELFWRAP_FLAGS =
640 ELFWRAP_FLAGS64 =      -64

642 #
643 # Various mapfiles that are used throughout the build, and delivered to

```

```

644 # /usr/lib/ld.
645 #
646 MAPFILE.NED_i386 = $(SRC)/common/mapfiles/common/map.noexdata
647 MAPFILE.NED_sparc =
648 MAPFILE.NED = $(MAPFILE.NED_$(MACH))
649 MAPFILE.PGA = $(SRC)/common/mapfiles/common/map.pagealign
650 MAPFILE.NES = $(SRC)/common/mapfiles/common/map.noexstk
651 MAPFILE.FLT = $(SRC)/common/mapfiles/common/map.filter
652 MAPFILE.LEX = $(SRC)/common/mapfiles/common/map.lex.yy

654 #
655 # Generated mapfiles that are compiler specific, and used throughout the
656 # build. These mapfiles are not delivered in /usr/lib/ld.
657 #
658 MAPFILE.NGB_sparc= $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
659 $(__GNUCC64)MAPFILE.NGB_sparc= \
660 $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
661 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
662 $(__GNUCC64)MAPFILE.NGB_sparcv9= \
663 $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
664 MAPFILE.NGB_i386= $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
665 $(__GNUCC64)MAPFILE.NGB_i386= \
666 $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
667 MAPFILE.NGB_amd64= $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
668 $(__GNUCC64)MAPFILE.NGB_amd64= \
669 $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
670 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH))

672 #
673 # A generic interface mapfile name, used by various dynamic objects to define
674 # the interfaces and interposers the object must export.
675 #
676 MAPFILE.INT = mapfile-intf

678 #
679 # LDLIBS32 and LDLIBS64 can be set in the environment to override the following
680 # assignments.
681 #
682 # These environment settings make sure that no libraries are searched outside
683 # of the local workspace proto area:
684 # LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
685 # LDLIBS64=-YP,$ROOT/lib:$MACH64:$ROOT/usr/lib:$MACH64
686 #
687 LDLIBS32 = $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
688 LDLIBS32 += $(ADJUNCT_PROTO:%=-L%/usr/lib -L%/lib)
689 LDLIBS.cmd = $(LDLIBS32)
690 LDLIBS.lib = $(LDLIBS32)

692 LDLIBS64 = $(ENVLDLIBS1:%=/%$(MACH64)) \
693 $(ENVLDLIBS2:%=/%$(MACH64)) \
694 $(ENVLDLIBS3:%=/%$(MACH64))
695 LDLIBS64 += $(ADJUNCT_PROTO:%=-L%/usr/lib/$MACH64 -L%/lib/$MACH64)

697 #
698 # Define compilation macros.
699 #
700 COMPILE.c= $(CC) $(CFLAGS) $(CPPFLAGS) -c
701 COMPILE64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) -c
702 COMPILE.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
703 COMPILE64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
704 COMPILE.s= $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
705 COMPILE64.s= $(AS) $(ASFLAGS) $(MACH64)_AS_XARCH $(AS_CPPFLAGS)
706 COMPILE.d= $(DTRACE) -G -32
707 COMPILE64.d= $(DTRACE) -G -64
708 COMPILE.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
709 COMPILE64.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

```

```

711 CLASSPATH= .
712 COMPILE.java= $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

714 #
715 # Link time macros
716 #
717 CCNEEDED = -lC
718 CCEXTNEEDED = -lCrun -lCstd
719 $(__GNUCC)CCNEEDED = -L$(GCCLIBDIR) -lstdc++ -lgcc_s
720 $(__GNUCC)CCEXTNEEDED = $(CCNEEDED)

722 LINK.c= $(CC) $(CFLAGS) $(CPPFLAGS) $(LDLFLAGS)
723 LINK64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDLFLAGS)
724 NORUNPATH= -norunpath -nolib
725 LINK.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
726 $(LDLFLAGS) $(CCNEEDED)
727 LINK64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
728 $(LDLFLAGS) $(CCNEEDED)

730 #
731 # lint macros
732 #
733 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
734 # ON is built with a version of lint that has the fix for 4484186.
735 #
736 ALWAYS_LINT_DEFS = -errtags=yes -s
737 ALWAYS_LINT_DEFS += -erroff=E_PTRDIFF_OVERFLOW
738 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_NARROW_CONV
739 ALWAYS_LINT_DEFS += -U__PRAGMA_REDEFINE_EXTNAME
740 ALWAYS_LINT_DEFS += $(C99LMODE)
741 ALWAYS_LINT_DEFS += -errsecurity=$(SECLEVEL)
742 ALWAYS_LINT_DEFS += -erroff=E_SEC_CREAT_WITHOUT_EXCL
743 ALWAYS_LINT_DEFS += -erroff=E_SEC_FORBIDDEN_WARN_CREAT
744 # XX64 -- really only needed for amd64 lint
745 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_INT_TO_SMALL_INT
746 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_CONST_TO_SMALL_INT
747 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_TO_SMALL_INT
748 ALWAYS_LINT_DEFS += -erroff=E_CAST_TO_PTR_FROM_INT
749 ALWAYS_LINT_DEFS += -erroff=E_COMP_INT_WITH_LARGE_INT
750 ALWAYS_LINT_DEFS += -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
751 ALWAYS_LINT_DEFS += -erroff=E_PASS_INT_TO_SMALL_INT
752 ALWAYS_LINT_DEFS += -erroff=E_PTR_CONV_LOSES_BITS

754 # This forces lint to pick up note.h and sys/note.h from Devpro rather than
755 # from the proto area. The note.h that ON delivers would disable NOTE().
756 ONLY_LINT_DEFS = -I$(SPRO_VROOT)/prod/include/lint

758 SECLEVEL= core
759 LINT.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
760 $(ALWAYS_LINT_DEFS)
761 LINT64.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
762 $(ALWAYS_LINT_DEFS)
763 LINT.s= $(LINT.c)

765 # For some future builds, NATIVE_MACH and MACH might be different.
766 # Therefore, NATIVE_MACH needs to be redefined in the
767 # environment as 'uname -p' to override this macro.
768 #
769 # For now at least, we cross-compile amd64 on i386 machines.
770 NATIVE_MACH= $(MACH:amd64=i386)

772 # Define native compilation macros
773 #

775 # Base directory where compilers are loaded.

```

```

776 # Defined here so it can be overridden by developer.
777 #
778 SPRO_ROOT=          $(BUILD_TOOLS)/SUNWspro
779 SPRO_VROOT=        $(SPRO_ROOT)/SS12
780 GNU_ROOT=          /usr

782 $(__GNUC)PRIMARY_CC= gcc4,$(GCC_ROOT)/bin/gcc,gnu
783 $(__SUNC)PRIMARY_CC= studio12,$(SPRO_VROOT)/bin/cc,sun
784 $(__GNUC)PRIMARY_CCC= gcc4,$(GCC_ROOT)/bin/g++,gnu
785 $(__SUNC)PRIMARY_CCC= studio12,$(SPRO_VROOT)/bin/CC,sun

787 CW_CC_COMPILERS=    $(PRIMARY_CC:%--primary %) $(SHADOW_CCS:%--shadow %)
788 CW_CCC_COMPILERS=   $(PRIMARY_CCC:%--primary %) $(SHADOW_CCCS:%--shadow %)

791 #endif /* ! codereview */
792 # Till SS12ul formally becomes the NV CBE, LINT is hard
793 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
794 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
795 # i386_LINT, amd64_LINT.
796 # Reset them when SS12ul is rolled out.
797 #

799 # Specify platform compiler versions for languages
800 # that we use (currently only c and c++).
801 #
802 CW=                  $(ONBLD_TOOLS)/bin/$(MACH)/cw

804 sparc_CC=           $(CW) $(CW_CC_COMPILERS) --
805 sparc_CCC=          $(CW) -C $(CW_CCC_COMPILERS) --
806 sparc_CC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
807 $(__GNUC)sparc_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
808 sparc_CCC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
809 $(__GNUC)sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
810 sparc_CPP=          /usr/ccs/lib/cpp
811 sparc_AS=           /usr/ccs/bin/as -xregsym=no
812 sparc_LD=           /usr/ccs/bin/ld
813 sparc_LINT=         $(SPRO_ROOT)/sunstudio12.1/bin/lint

811 sparcv9_CC=        $(CW) $(CW_CC_COMPILERS) --
812 sparcv9_CCC=       $(CW) -C $(CW_CCC_COMPILERS) --
813 sparcv9_CC=        $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
814 $(__GNUC64)sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
815 sparcv9_CCC=       $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
816 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
817 sparcv9_CPP=       /usr/ccs/lib/cpp
818 sparcv9_AS=        /usr/ccs/bin/as -xregsym=no
819 sparcv9_LD=        /usr/ccs/bin/ld
820 sparcv9_LINT=      $(SPRO_ROOT)/sunstudio12.1/bin/lint

818 i386_CC=           $(CW) $(CW_CC_COMPILERS) --
819 i386_CCC=          $(CW) -C $(CW_CCC_COMPILERS) --
820 i386_CC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
821 $(__GNUC)i386_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
822 i386_CCC=         $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
823 $(__GNUC)i386_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
824 i386_CPP=          /usr/ccs/lib/cpp
825 i386_AS=           /usr/ccs/bin/as
826 $(__GNUC)i386_AS=  $(ONBLD_TOOLS)/bin/$(MACH)/aw
827 i386_LD=           /usr/ccs/bin/ld
828 i386_LINT=         $(SPRO_ROOT)/sunstudio12.1/bin/lint

826 amd64_CC=          $(CW) $(CW_CC_COMPILERS) --
827 amd64_CCC=         $(CW) -C $(CW_CCC_COMPILERS) --
828 amd64_CC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
829 $(__GNUC64)amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc

```

```

814 amd64_CCC=         $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
815 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
828 amd64_CPP=        /usr/ccs/lib/cpp
829 amd64_AS=         $(ONBLD_TOOLS)/bin/$(MACH)/aw
830 amd64_LD=         /usr/ccs/bin/ld
831 amd64_LINT=       $(SPRO_ROOT)/sunstudio12.1/bin/lint

833 NATIVECC=         $( $(NATIVE_MACH)_CC )
834 NATIVECCC=        $( $(NATIVE_MACH)_CCC )
835 NATIVECPP=        $( $(NATIVE_MACH)_CPP )
836 NATIVEAS=         $( $(NATIVE_MACH)_AS )
837 NATIVELD=         $( $(NATIVE_MACH)_LD )
838 NATIVELINT=       $( $(NATIVE_MACH)_LINT )

840 #
841 # Makefile.master.64 overrides these settings
842 #
843 CC=                $(NATIVECC)
844 CCC=               $(NATIVECCC)
845 CPP=               $(NATIVECPP)
846 AS=                $(NATIVEAS)
847 LD=                $(NATIVELD)
848 LINT=              $(NATIVELINT)

838 # The real compilers used for this build
839 CW_CC_CMD=         $(CC) -_compiler
840 CW_CCC_CMD=        $(CCC) -_compiler
841 REAL_CC=           $(CW_CC_CMD:sh)
842 REAL_CCC=          $(CW_CCC_CMD:sh)

850 # Pass -Y flag to cpp (method of which is release-dependent)
851 CCYFLAG=          -Y I,

853 BDIRECT=          -Bdirect
854 BDYNAMIC=         -Bdynamic
855 BLOCAL=           -Blocal
856 BNODIRECT=        -Bnodirect
857 BREDUCE=          -Breduce
858 BSTATIC=          -Bstatic

860 ZDEFS=            -zdefs
861 ZDIRECT=          -zdirect
862 ZIGNORE=          -zignore
863 ZINITFIRST=       -zinitfirst
864 ZINTERPOSE=       -zinterpose
865 ZLAZYLOAD=        -zlazyload
866 ZLOADFLTR=       -zloadfltr
867 ZMULDEFS=         -zmuldefs
868 ZNODEFAULTLIB=   -znodefaultlib
869 ZNODEFS=          -znodefs
870 ZNODELETE=        -znodelete
871 ZNODLOPEN=        -znodlopen
872 ZNODUMP=          -znodump
873 ZNOLAZYLOAD=      -znolazyload
874 ZNOLDYNSYM=       -znoldynsym
875 ZNORELOC=         -znoreloc
876 ZNOVERSION=       -znoverion
877 ZRECORD=          -zrecord
878 ZREDLOCSYM=       -zredlocsymb
879 ZTEXT=            -ztext
880 ZVERBOSE=         -zverbose

882 GSHARED=          -G
883 CCMT=             -mt

885 # Handle different PIC models on different ISAs

```

```

886 # (May be overridden by lower-level Makefiles)

888 sparc_C_PICFLAGS =      -K pic
889 sparcv9_C_PICFLAGS =    -K pic
890 i386_C_PICFLAGS =        -K pic
891 amd64_C_PICFLAGS =       -K pic
892 C_PICFLAGS =              $( $(MACH)_C_PICFLAGS )
893 C_PICFLAGS64 =           $( $(MACH64)_C_PICFLAGS )

895 sparc_C_BIGPICFLAGS =    -K PIC
896 sparcv9_C_BIGPICFLAGS = -K PIC
897 i386_C_BIGPICFLAGS =    -K PIC
898 amd64_C_BIGPICFLAGS =    -K PIC
899 C_BIGPICFLAGS =          $( $(MACH)_C_BIGPICFLAGS )
900 C_BIGPICFLAGS64 =        $( $(MACH64)_C_BIGPICFLAGS )

902 # CC requires there to be no space between '-K' and 'pic' or 'PIC'.
903 sparc_CC_PICFLAGS =      -Kpic
904 sparcv9_CC_PICFLAGS =    -KPIC
905 i386_CC_PICFLAGS =       -Kpic
906 amd64_CC_PICFLAGS =      -Kpic
907 CC_PICFLAGS =            $( $(MACH)_CC_PICFLAGS )
908 CC_PICFLAGS64 =          $( $(MACH64)_CC_PICFLAGS )

910 AS_PICFLAGS=             $(C_PICFLAGS)
911 AS_BIGPICFLAGS=          $(C_BIGPICFLAGS)

913 #
914 # Default label for CTF sections
915 #
916 CTFCVTFLAGS=             -i -L VERSION

918 #
919 # Override to pass module-specific flags to ctfmerge.  Currently used only by
920 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
921 # stripping.
922 #
923 CTFMRGFLAGS=

925 CTFCONVERT_O             = $(CTFCONVERT) $(CTFCVTFLAGS) $@

927 # Rules (normally from make.rules) and macros which are used for post
928 # processing files.  Normally, these do stripping of the comment section
929 # automatically.
930 #   RELEASE_CM:           Should be edited to reflect the release.
931 #   POST_PROCESS_O:       Post-processing for '.o' files.
932 #   POST_PROCESS_A:       Post-processing for '.a' files (currently null).
933 #   POST_PROCESS_SO:      Post-processing for '.so' files.
934 #   POST_PROCESS:         Post-processing for executable files (no suffix).
935 # Note that these macros are not completely generalized as they are to be
936 # used with the file name to be processed following.
937 #
938 # It is left as an exercise to Release Engineering to embellish the generation
939 # of the release comment string.
940 #
941 #   If this is a standard development build:
942 #     compress the comment section (mcs -c)
943 #     add the standard comment (mcs -a $(RELEASE_CM))
944 #     add the development specific comment (mcs -a $(DEV_CM))
945 #
946 #   If this is an installation build:
947 #     delete the comment section (mcs -d)
948 #     add the standard comment (mcs -a $(RELEASE_CM))
949 #     add the development specific comment (mcs -a $(DEV_CM))
950 #
951 #   If this is an release build:

```

```

952 #           delete the comment section (mcs -d)
953 #           add the standard comment (mcs -a $(RELEASE_CM))
954 #
955 # The following list of macros are used in the definition of RELEASE_CM
956 # which is used to label all binaries in the build:
957 #
958 #   RELEASE           Specific release of the build, eg: 5.2
959 #   RELEASE_MAJOR     Major version number part of $(RELEASE)
960 #   RELEASE_MINOR     Minor version number part of $(RELEASE)
961 #   VERSION           Version of the build (alpha, beta, Generic)
962 #   PATCHID           If this is a patch this value should contain
963 #                     the patchid value (eg: "Generic 100832-01"), otherwise
964 #                     it will be set to $(VERSION)
965 #   RELEASE_DATE      Date of the Release Build
966 #   PATCH_DATE        Date the patch was created, if this is blank it
967 #                     will default to the RELEASE_DATE
968 #
969 RELEASE_MAJOR= 5
970 RELEASE_MINOR= 11
971 RELEASE=       $(RELEASE_MAJOR).$(RELEASE_MINOR)
972 VERSION=       SunOS Development
973 PATCHID=       $(VERSION)
974 RELEASE_DATE=  release date not set
975 PATCH_DATE=    $(RELEASE_DATE)
976 RELEASE_CM=    "@$( POUND_SIGN )SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
977 DEV_CM=        "@$( POUND_SIGN )SunOS Internal Development: non-nightly build"

979 PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
980 $(RELEASE_BUILD)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM)

982 STRIP_STABS=    $(STRIP) -x $@
983 $(SRCDBGBLD)STRIP_STABS= :

985 POST_PROCESS_O=
986 POST_PROCESS_A=
987 POST_PROCESS_SO= $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
988                  $(ELFSIGN_OBJECT)
989 POST_PROCESS=    $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
990                  $(ELFSIGN_OBJECT)

992 #
993 # chk4ubin is a tool that inspects a module for a symbol table
994 # ELF section size which can trigger an OBP bug on older platforms.
995 # This problem affects only specific sun4u bootable modules.
996 #
997 CHK4UBIN=       $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
998 CHK4UBINFLAGS=
999 CHK4UBINARY=    $(CHK4UBIN) $(CHK4UBINFLAGS) $@

1001 #
1002 # PKGARCHIVE specifies the default location where packages should be
1003 # placed if built.
1004 #
1005 $(RELEASE_BUILD)PKGARCHIVESUFFIX= -nd
1006 PKGARCHIVE=$(SRC)/../../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)

1008 #
1009 # The repositories will be created with these publisher settings.  To
1010 # update an image to the resulting repositories, this must match the
1011 # publisher name provided to "pkg set-publisher."
1012 #
1013 PKGPUBLISHER_REDIST= on-nightly
1014 PKGPUBLISHER_NONREDIST= on-extra

1016 #           Default build rules which perform comment section post-processing.
1017 #

```

```

1018 .c:
1019     $(LINK.c) -o $@ $< $(LDLIBS)
1020     $(POST_PROCESS)
1021 .c.o:
1022     $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1023     $(POST_PROCESS_O)
1024 .c.a:
1025     $(COMPILE.c) -o $% $<
1026     $(PROCESS_COMMENT) $%
1027     $(AR) $(ARFLAGS) $@ $%
1028     $(RM) $%
1029 .s.o:
1030     $(COMPILE.s) -o $@ $<
1031     $(POST_PROCESS_O)
1032 .s.a:
1033     $(COMPILE.s) -o $% $<
1034     $(PROCESS_COMMENT) $%
1035     $(AR) $(ARFLAGS) $@ $%
1036     $(RM) $%
1037 .cc:
1038     $(LINK.cc) -o $@ $< $(LDLIBS)
1039     $(POST_PROCESS)
1040 .cc.o:
1041     $(COMPILE.cc) $(OUTPUT_OPTION) $<
1042     $(POST_PROCESS_O)
1043 .cc.a:
1044     $(COMPILE.cc) -o $% $<
1045     $(AR) $(ARFLAGS) $@ $%
1046     $(PROCESS_COMMENT) $%
1047     $(RM) $%
1048 .y:
1049     $(YACC.y) $<
1050     $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1051     $(POST_PROCESS)
1052     $(RM) y.tab.c
1053 .y.o:
1054     $(YACC.y) $<
1055     $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1056     $(POST_PROCESS_O)
1057     $(RM) y.tab.c
1058 .l:
1059     $(RM) $*.c
1060     $(LEX.l) $< > $*.c
1061     $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1062     $(POST_PROCESS)
1063     $(RM) $*.c
1064 .l.o:
1065     $(RM) $*.c
1066     $(LEX.l) $< > $*.c
1067     $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1068     $(POST_PROCESS_O)
1069     $(RM) $*.c

1071 .bin.o:
1072     $(COMPILE.b) -o $@ $<
1073     $(POST_PROCESS_O)

1075 .java.class:
1076     $(COMPILE.java) $<

1078 # Bourne and Korn shell script message catalog build rules.
1079 # We extract all gettext strings with sed(1) (being careful to permit
1080 # multiple gettext strings on the same line), weed out the dups, and
1081 # build the catalogue with awk(1).

1083 .sh.po .ksh.po:

```

```

1084     $(SED) -n -e ":a" \
1085     -e "h" \
1086     -e "s/. *gettext *\([^\"]*\)*\.\.*/\1/p" \
1087     -e "x" \
1088     -e "s/\([^\"]*\)gettext *\([^\"]*\)*\.\.*/\1\2/" \
1089     -e "t a" \
1090     $< | sort -u | $(AWK) '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1092 #
1093 # Python and Perl executable and message catalog build rules.
1094 #
1095 .SUFFIXES: .pl .pm .py .pyc

1097 .pl:
1098     $(RM) $@;
1099     $(SED) -e "s@TEXT_DOMAIN@"$(TEXT_DOMAIN)"@" $< > $@;
1100     $(CHMOD) +x $@

1102 .py:
1103     $(RM) $@; $(SED) -e "1s:^(\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $(CHMOD)

1105 .py.pyc:
1106     $(RM) $@
1107     $(PYTHON) -mpy_compile $<
1108     @[ $(<)c = $@ ] || $(MV) $(<)c $@

1110 .py.po:
1111     $(GNUXGETTEXT) $(GNUXGETTEXTFLAGS) -d $(<F:%.py=) $< ;

1113 .pl.po .pm.po:
1114     $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $< ;
1115     $(RM) $@ ;
1116     $(SED) "/^domain/d" < $(<F).po > $@ ;
1117     $(RM) $(<F).po

1119 #
1120 # When using xgettext, we want messages to go to the default domain,
1121 # rather than the specified one. This special version of the
1122 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1123 # causing xgettext to put all messages into the default domain.
1124 #
1125 CPPFORPO=$(COMPILE.cpp:"$(TEXT_DOMAIN)"=TEXT_DOMAIN)

1127 .c.i:
1128     $(CPPFORPO) $< > $@

1130 .h.i:
1131     $(CPPFORPO) $< > $@

1133 .y.i:
1134     $(YACC) -d $<
1135     $(CPPFORPO) y.tab.c > $@
1136     $(RM) y.tab.c

1138 .l.i:
1139     $(LEX) $<
1140     $(CPPFORPO) lex.yy.c > $@
1141     $(RM) lex.yy.c

1143 .c.po:
1144     $(CPPFORPO) $< > $<.i
1145     $(BUILD.po)

1147 .cc.po:
1148     $(CPPFORPO) $< > $<.i
1149     $(BUILD.po)

```

```
1151 .y.po:
1152     $(YACC) -d $<
1153     $(CPPFORPO) y.tab.c > $<.i
1154     $(BUILD.po)
1155     $(RM) y.tab.c

1157 .l.po:
1158     $(LEX) $<
1159     $(CPPFORPO) lex.yy.c > $<.i
1160     $(BUILD.po)
1161     $(RM) lex.yy.c

1163 #
1164 # Rules to perform stylistic checks
1165 #
1166 .SUFFIXES: .x .xml .check .xmlchk

1168 .h.check:
1169     $(DOT_H_CHECK)

1171 .x.check:
1172     $(DOT_X_CHECK)

1174 .xml.xmlchk:
1175     $(MANIFEST_CHECK)

1177 #
1178 # Include rules to render automated sccs get rules "safe".
1179 #
1180 include $(SRC)/Makefile.noget
```

new/usr/src/Makefile.master.64

1

```
*****
2199 Thu Feb 22 01:20:09 2018
new/usr/src/Makefile.master.64
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1997, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
24 #
25 #
26 # rebind basic build macros to 64-bit versions
27 #
28 CFLAGS= $(CFLAGS64)
29 CCFLAGS= $(CCFLAGS64)
30 COPTFLAG= $(COPTFLAG64)
31 COPTFLAG= $(COPTFLAG64)
32 COMPILE.c= $(COMPILE64.c)
33 COMPILE.cc= $(COMPILE64.cc)
34 COMPILE.s= $(COMPILE64.s)
35 COMPILE.d= $(COMPILE64.d)
36 COMPILE.b= $(COMPILE64.b)
37 LINK.c= $(LINK64.c)
38 LINK.cc= $(LINK64.cc)
39 LINT.c= $(LINT64.c)
40 #
41 CTF_FLAGS= $(CTF_FLAGS_64)
42 #
43 OFFSETS_CREATE= $(OFFSETS_CREATE64)
44 #
45 #
46 # Moved these up to the top level here, so they can be overridden
47 #
48 LDLIBS.cmd = $(LDLIBS64)
49 LDLIBS.lib = $(LDLIBS64)
50 #
51 #
52 # Override Makefile.master's settings of GCCLIBDIR
53 #
54 GCCLIBDIR= $(GCCLIBDIR64)
```

new/usr/src/Makefile.master.64

2

```
56 #
57 # Override Makefile.master's settings of CC, CCC, AS and LINT
58 #
59 CC= $(MACH64)_CC
60 CCC= $(MACH64)_CCC
61 CPP= $(MACH64)_CPP
62 AS= $(MACH64)_AS
63 LD= $(MACH64)_LD
64 LINT= $(MACH64)_LINT
65 #
66 REAL_CC= $(CW_CC_CMD:sh)
67 REAL_CCC= $(CW_CCC_CMD:sh)
68 #
69 BUILD.SO= $(CC) $(CFLAGS) -o $@ $(GSHARED) $(DYNFLAGS) \
70 $(PICS) $(EXTPICS) -L $(ROOTLIBDIR64) $(LDLIBS)
71 #
72 MAPFILECLASS= -64
73 #
74 CCNEEDED = -lCrun
75 $(__GNUC64)CCNEEDED = -L$(GCCLIBDIR) -lstdc++
76 $(__GNUC64)CCNEEDED += -lgcc_s
77 #
78 BUILDCCC.SO= $(CCC) $(CCFLAGS) -o $@ $(GSHARED) $(DYNFLAGS) \
79 $(PICS) $(EXTPICS) $(LDLIBS) $(CCNEEDED)
80 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH64))
```



new/usr/src/cmd/dtrace/test/tst/common/Makefile

1

```
*****
5441 Thu Feb 22 01:20:11 2018
new/usr/src/cmd/dtrace/test/tst/common/Makefile
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #
27 #
28 # Copyright (c) 2012 by Delphix. All rights reserved.
29 # Copyright (c) 2013, Joyent, Inc. All rights reserved.
30 # Copyright 2015 Nexenta Systems, Inc. All rights reserved.
31 #
32 #
33 #
34 # WARNING: Do not include Makefile.ctf here. That will cause tests to
35 # break.
36 #
37 #
38 include $(SRC)/Makefile.master
39 include ../Makefile.com
40 #
41 SNOOPDIR = $(SRC)/cmd/cmd-inet/usr/sbin/snoop
42 SNOOPOBJS = nfs4_xdr.o
43 SNOOPSRCS = ${SNOOPOBJS:%.o=%.c}
44 CLOBBERFILES += nfs/${SNOOPOBJS}
45 #
46 RPCSVCDIR = $(SRC)/head/rpcsvc
47 RPCSVCOBJS = nfs_prot.o
48 RPCSVCSRCS = ${RPCSVCOBJS:%o=%c}
49 CLOBBERFILES += nfs/${RPCSVCOBJS} ${RPCSVCDIR}/${RPCSVCSRCS}
50 CLOBBERFILES += usdt/forcker.h usdt/lazyprobe.h
51 #
52 fasttrap/tst.fasttrap.exe := LDLIBS += -ldtrace
53 fasttrap/tst.stack.exe := LDLIBS += -ldtrace
```

new/usr/src/cmd/dtrace/test/tst/common/Makefile

2

```
55 sysevent/tst.post.exe := LDLIBS += -lsysevent
56 sysevent/tst.post_chan.exe := LDLIBS += -lsysevent
57 #
58 ustack/tst.bigstack.exe := COPTFLAG += -x01
59 #
60 GCC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -gcc
61 #
62 nfs/%.o: $(SNOOPDIR)/%.c
63 $(COMPILE.c) -o $@ $< -I$(SNOOPDIR)
64 $(POST_PROCESS_O)
65 nfs/tst.call.exe: nfs/tst.call.o nfs/${SNOOPOBJS}
66 $(LINK.c) -o $@ nfs/tst.call.o nfs/${SNOOPOBJS} $(LDLIBS) -lnsl
67 $(POST_PROCESS) ; $(STRIP_STABS)
68 $(RPCSVCDIR)/%.c: $(RPCSVCDIR)/%.x
69 $(RPCGEN) -Cc $< > $@
70 nfs/${RPCSVCOBJS}: $(RPCSVCDIR)/$(RPCSVCSRCS)
71 $(COMPILE.c) -o $@ $(RPCSVCDIR)/$(RPCSVCSRCS)
72 $(POST_PROCESS_O)
73 nfs/tst.call3.exe: nfs/tst.call3.o nfs/${RPCSVCOBJS}
74 $(LINK.c) -o $@ nfs/tst.call3.o nfs/${RPCSVCOBJS} \
75 $(LDLIBS) -lnsl -lrpcsvc
76 $(POST_PROCESS) ; $(STRIP_STABS)
77 #
78 pid/tst.gcc.exe: pid/tst.gcc.c
79 $(GCC) -o pid/tst.gcc.exe pid/tst.gcc.c $(LDLFLAGS) $(LDLIBS)
80 $(POST_PROCESS) ; $(STRIP_STABS)
81 #
82 json/usdt.o: json/usdt.h
83 #
84 json/usdt.h: json/usdt.d
85 $(DTRACE) -h -s json/usdt.d -o json/usdt.h
86 #
87 CLOBBERFILES += json/usdt.h
88 #
89 json/usdt.o: json/usdt.d json/tst.usdt.o
90 $(COMPILE.d) -o json/usdt.o -s json/usdt.d json/tst.usdt.o
91 #
92 json/tst.usdt.exe: json/tst.usdt.o json/usdt.o
93 $(LINK.c) -o json/tst.usdt.exe json/tst.usdt.o json/usdt.o $(LDLIBS)
94 $(POST_PROCESS) ; $(STRIP_STABS)
95 #
96 # Tests that use the next three programs rely on the binaries having
97 # valid CTF data.
98 #
99 uctf/tst.aouttype.exe: uctf/tst.aouttype.c
100 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.aouttype.o uctf/tst.aouttype.c
101 $(CTFCONVERT) -i -L VERSION uctf/tst.aouttype.o
102 $(LINK.c) -o uctf/tst.aouttype.exe uctf/tst.aouttype.o $(LDLIBS)
103 $(CTFMERGE) -L VERSION -o $@ uctf/tst.aouttype.o
104 $(POST_PROCESS) ; $(STRIP_STABS)
105 #
106 uctf/tst.chasestrings.exe: uctf/tst.chasestrings.c
107 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.chasestrings.o uctf/tst.chasestrin
108 $(CTFCONVERT) -i -L VERSION uctf/tst.chasestrings.o
109 $(LINK.c) -o uctf/tst.chasestrings.exe uctf/tst.chasestrings.o $(LDLIBS)
110 $(CTFMERGE) -L VERSION -o $@ uctf/tst.chasestrings.o
111 $(POST_PROCESS) ; $(STRIP_STABS)
112 #
113 uctf/tst.printtype.exe: uctf/tst.printtype.c
114 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.printtype.o uctf/tst.printtype.c
115 $(CTFCONVERT) -i -L VERSION uctf/tst.printtype.o
116 $(LINK.c) -o uctf/tst.printtype.exe uctf/tst.printtype.o $(LDLIBS)
117 $(CTFMERGE) -L VERSION -o $@ uctf/tst.printtype.o
118 $(POST_PROCESS) ; $(STRIP_STABS)
```

```
115 #
116 # This program should never have any ctf data in it.
117 #
118 uctf/tst.libtype.exe:
119     $(LINK.c) -o uctf/tst.libtype.exe uctf/tst.libtype.c $(LDLIBS)
120     $(POST_PROCESS) ; $(STRIP_STABS)

122 usdt/tst.args.exe: usdt/tst.args.o usdt/args.o
123     $(LINK.c) -o usdt/tst.args.exe usdt/tst.args.o usdt/args.o $(LDLIBS)
124     $(POST_PROCESS) ; $(STRIP_STABS)

126 usdt/args.o: usdt/args.d usdt/tst.args.o
127     $(COMPILE.d) -o usdt/args.o -s usdt/args.d usdt/tst.args.o

129 usdt/tst.argmap.exe: usdt/tst.argmap.o usdt/argmap.o
130     $(LINK.c) -o usdt/tst.argmap.exe \
131         usdt/tst.argmap.o usdt/argmap.o $(LDLIBS)
132     $(POST_PROCESS) ; $(STRIP_STABS)

134 usdt/argmap.o: usdt/argmap.d usdt/tst.argmap.o
135     $(COMPILE.d) -o usdt/argmap.o -s usdt/argmap.d usdt/tst.argmap.o

137 usdt/tst.forker.exe: usdt/tst.forker.o usdt/forker.o
138     $(LINK.c) -o usdt/tst.forker.exe \
139         usdt/tst.forker.o usdt/forker.o $(LDLIBS)
140     $(POST_PROCESS) ; $(STRIP_STABS)

142 usdt/forker.o: usdt/forker.d usdt/tst.forker.o
143     $(COMPILE.d) -o usdt/forker.o -s usdt/forker.d usdt/tst.forker.o

145 usdt/tst.forker.o: usdt/forker.h

147 usdt/forker.h: usdt/forker.d
148     $(DTRACE) -h -s usdt/forker.d -o usdt/forker.h

150 usdt/tst.lazyprobe.exe: usdt/tst.lazyprobe.o usdt/lazyprobe.o
151     $(LINK.c) -o usdt/tst.lazyprobe.exe \
152         usdt/tst.lazyprobe.o usdt/lazyprobe.o $(LDLIBS)
153     $(POST_PROCESS) ; $(STRIP_STABS)

155 usdt/lazyprobe.o: usdt/lazyprobe.d usdt/tst.lazyprobe.o
156     $(COMPILE.d) -xlazyload -o usdt/lazyprobe.o \
157         -s usdt/lazyprobe.d usdt/tst.lazyprobe.o

159 usdt/tst.lazyprobe.o: usdt/lazyprobe.h

161 usdt/lazyprobe.h: usdt/lazyprobe.d
162     $(DTRACE) -h -s usdt/lazyprobe.d -o usdt/lazyprobe.h

164 SUBDIRS = java_api
165 include ../../Makefile.subdirs
```

```

*****
4425 Thu Feb 22 01:20:12 2018
new/usr/src/common/mapfiles/gen/Makefile
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
26 #
27 #
28 include $(SRC)/Makefile.master
29 #
30 #
31 $(__GNUC)FILES= $(MACH)_gcc_map.noexeglobs
32 $(__SUNC)FILES= $(MACH)_cc_map.noexeglobs
33 #
34 $(__GNUC)$(BUILD64)FILES += $(MACH64)_gcc_map.noexeglobs
35 $(__SUNC)$(BUILD64)FILES += $(MACH64)_cc_map.noexeglobs
36 FILES= \
37     $(MACH)_cc_map.noexeglobs \
38     $(MACH)_gcc_map.noexeglobs
39 #
40 $(BUILD64)FILES += \
41     $(MACH64)_cc_map.noexeglobs \
42     $(MACH64)_gcc_map.noexeglobs
43 #
44 SYMS1= syms.1
45 SYMS2= syms.2
46 MAIN1= main.1
47 MAIN2= main.2
48 #
49 TEMPLATE1= map.noexeglobs.1.template
50 TEMPLATE2= map.noexeglobs.2.template
51 #
52 all install: $(FILES)
53 #
54 lint:

```

```

49 clean:
50     $(RM) $(SYMS1) $(SYMS2) $(MAIN1) $(MAIN2)
51 #
52 clobber: clean
53     $(RM) $(FILES)
54 #
55 # A number of dynamic executables have their own definitions of interfaces that
56 # exist in system libraries. To prevent name-space pollution it is desirable
57 # to demote the interfaces within these executable to locals. However, various
58 # symbols defined by the compiler drivers crt/values files need to remain
59 # global in any dynamic object that includes these files. These symbols
60 # interpose on symbols in libc, or provide call backs for other system
61 # libraries. The various compiler drivers (cc and gcc), and the crt/values
62 # files that these drivers are configured to include, differ between the
63 # various compilation environments (platform, 32/64-bit). Therefore, the
64 # only means of creating a mapfile to demote symbols is to dynamically generate
65 # the mapfile for a specific compilation environment.
66 #
67 # Here, template mapfiles are used to generate a number of compilation specific
68 # mapfiles. These mapfiles are referenced by components of the build through
69 # the MAPFILE.NGB macro defined in Makefile.master. These dynamically created
70 # mapfiles are not delivered into the $ROOT area, and therefore are not
71 # delivered as packaged components of the OSNet.
72 #
73 $(MACH)_cc_map.noexeglobs := LINK = $(LINK.c)
74 $(MACH64)_cc_map.noexeglobs := LINK = $(LINK64.c)
75 #
76 $(MACH)_gcc_map.noexeglobs := CC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
77 $(MACH)_gcc_map.noexeglobs := LINK = $(LINK.c)
78 $(MACH64)_gcc_map.noexeglobs := CC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
79 $(MACH64)_gcc_map.noexeglobs := LINK = $(LINK64.c)
80 #
81 # This generic target creates two dynamic executables from an empty "main"
82 # program. These objects are not executed, but are analyzed to determine the
83 # global symbols each provides.
84 #
85 # The first executable demotes a family of known interfaces to local and allows
86 # all other symbol definitions to remain global. This executables provides the
87 # base for discovering all symbol definitions provided by the various
88 # compilation environments. The second executable demotes all symbols to
89 # locals. Within both executables, some symbols remain globals (_end, _etext,
90 # etc.) as the link-editor has special knowledge of these symbols and their
91 # expected visibility requirements. By inspecting the deferences between the
92 # global symbols within the two executables, a mapfile can be generated to
93 # ensure the symbols defined by the compilation environments files remain
94 # global.
95 #
96 %map.noexeglobs:main.c $(TEMPLATE1) $(TEMPLATE2)
97     $(LINK) -o $(MAIN1) -M$(TEMPLATE1) main.c
98     $(ELFDUMP) -s -N.dynsym $(MAIN1) | $(EGREP) "WEAK[GLOB] | \
99     $(GREP) -v UNDEF | $(AWK) '{print $$9}' | $(SORT) > $(SYMS1)
100     $(LINK) -o $(MAIN2) -M$(TEMPLATE2) main.c
101     $(ELFDUMP) -s -N.dynsym $(MAIN2) | $(EGREP) "WEAK[GLOB] | \
102     $(GREP) -v UNDEF | $(AWK) '{print $$9}' | $(SORT) > $(SYMS2)
103     $(ECHO) "# GENERATED FILE - DO NOT EDIT" > $@
104     $(GREP) MAP-HEAD $(TEMPLATE2) | \
105     $(SED) -e "s/ *# MAP-HEAD//" >> $@
106     $(DIFF) $(SYMS1) $(SYMS2) | $(GREP) "^<" | \
107     $(SED) -e "s/^< \(.*\)/ \1;/" >> $@
108     $(GREP) MAP-TAIL $(TEMPLATE2) | \
109     $(SED) -e "s/ *# MAP-TAIL//" >> $@
110     $(RM) $(SYMS1) $(SYMS2) $(MAIN1) $(MAIN2)

```

new/usr/src/lib/smbstrv/Makefile.targ

1

\*\*\*\*\*

1117 Thu Feb 22 01:20:13 2018

new/usr/src/lib/smbstrv/Makefile.targ

9128 cw(lonbld) should be able to run multiple shadows

9129 file-locking tests shouldn't build multiple source files in one compiler in

9130 DTrace tst.gcc.d isn't useful

9132 cw(lonbld) shouldn't shadow pure preprocessing

Reviewed by: Toomas Soome <tsoome@me.com>

Reviewed? by: Yuri Pankov <yuripv@yuripv.net>

Reviewed? by: Robert Mustacchi <rm@joyent.com>

Reviewed? by: Jason King <jason.king@joyent.com>

\*\*\*\*\*

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 #ident "%Z%M% %I% %E% SMI"

26 #
27 # Common targets for smbstrv Makefiles
28 #

30 _ndr.c: $(NDDLDIR)/%.ndl
31     $(NDRGEN) -Y "$(CC)" $<
32     $(NDRGEN) -Y $(CC) $<

33 pics/%.o: $(SRC)/common/smbstrv/%.c
34     $(COMPILE.c) -o $@ $<
35     $(POST_PROCESS_O)

37 .KEEP_STATE:

39 all: $(LIBS)

41 lint: lintcheck
```

new/usr/src/pkg/manifests/system-dtrace-tests.mf

1

```
*****
125305 Thu Feb 22 01:20:15 2018
new/usr/src/pkg/manifests/system-dtrace-tests.mf
9128 cw(lonbld) shouldn't be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012, 2016 by Delphix. All rights reserved.
25 #
26 #
27 set name=pkg.fmri value=pkg:/system/dtrace/tests@$(PKGVERS)
28 set name=pkg.description value="DTrace Test Suite Internal Distribution"
29 set name=pkg.summary value="DTrace Test Suite"
30 set name=info.classification \
31     value=org.opensolaris.category.2008:Development/System
32 set name=variant.arch value=$(ARCH)
33 dir path=opt/SUNWdtrt group=sys
34 dir path=opt/SUNWdtrt/bin
35 dir path=opt/SUNWdtrt/bin/$(ARCH32)
36 dir path=opt/SUNWdtrt/bin/$(ARCH64)
37 dir path=opt/SUNWdtrt/lib
38 dir path=opt/SUNWdtrt/lib/java
39 dir path=opt/SUNWdtrt/tst
40 dir path=opt/SUNWdtrt/tst/$(ARCH)
41 dir path=opt/SUNWdtrt/tst/$(ARCH)/arrays
42 $(i386_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/error
43 $(i386_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/funcs
44 dir path=opt/SUNWdtrt/tst/$(ARCH)/pid
45 $(sparc_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/usdt
46 dir path=opt/SUNWdtrt/tst/$(ARCH)/ustack
47 dir path=opt/SUNWdtrt/tst/common
48 dir path=opt/SUNWdtrt/tst/common/aggs
49 dir path=opt/SUNWdtrt/tst/common/arithmetic
50 dir path=opt/SUNWdtrt/tst/common/arrays
51 dir path=opt/SUNWdtrt/tst/common/assocs
52 dir path=opt/SUNWdtrt/tst/common/begin
53 dir path=opt/SUNWdtrt/tst/common/bitfields
54 dir path=opt/SUNWdtrt/tst/common/buffering
```

new/usr/src/pkg/manifests/system-dtrace-tests.mf

2

```
55 dir path=opt/SUNWdtrt/tst/common/builtinvar
56 dir path=opt/SUNWdtrt/tst/common/cg
57 dir path=opt/SUNWdtrt/tst/common/clauses
58 dir path=opt/SUNWdtrt/tst/common/cpc
59 dir path=opt/SUNWdtrt/tst/common/decls
60 dir path=opt/SUNWdtrt/tst/common/drops
61 dir path=opt/SUNWdtrt/tst/common/dtraceUtil
62 dir path=opt/SUNWdtrt/tst/common/end
63 dir path=opt/SUNWdtrt/tst/common/enum
64 dir path=opt/SUNWdtrt/tst/common/env
65 dir path=opt/SUNWdtrt/tst/common/error
66 dir path=opt/SUNWdtrt/tst/common/exit
67 dir path=opt/SUNWdtrt/tst/common/fbtprovider
68 dir path=opt/SUNWdtrt/tst/common/funcs
69 dir path=opt/SUNWdtrt/tst/common/grammar
70 dir path=opt/SUNWdtrt/tst/common/include
71 dir path=opt/SUNWdtrt/tst/common/inline
72 dir path=opt/SUNWdtrt/tst/common/io
73 dir path=opt/SUNWdtrt/tst/common/ip
74 dir path=opt/SUNWdtrt/tst/common/java_api
75 dir path=opt/SUNWdtrt/tst/common/json
76 dir path=opt/SUNWdtrt/tst/common/lexer
77 dir path=opt/SUNWdtrt/tst/common/llquantize
78 dir path=opt/SUNWdtrt/tst/common/mdb
79 dir path=opt/SUNWdtrt/tst/common/mib
80 dir path=opt/SUNWdtrt/tst/common/misc
81 dir path=opt/SUNWdtrt/tst/common/multiaggs
82 dir path=opt/SUNWdtrt/tst/common/nfs
83 dir path=opt/SUNWdtrt/tst/common/offsetof
84 dir path=opt/SUNWdtrt/tst/common/operators
85 dir path=opt/SUNWdtrt/tst/common/pid
86 dir path=opt/SUNWdtrt/tst/common/plockstat
87 dir path=opt/SUNWdtrt/tst/common/pointers
88 dir path=opt/SUNWdtrt/tst/common/pragma
89 dir path=opt/SUNWdtrt/tst/common/predicates
90 dir path=opt/SUNWdtrt/tst/common/preprocessor
91 dir path=opt/SUNWdtrt/tst/common/print
92 dir path=opt/SUNWdtrt/tst/common/printa
93 dir path=opt/SUNWdtrt/tst/common/printf
94 dir path=opt/SUNWdtrt/tst/common/privs
95 dir path=opt/SUNWdtrt/tst/common/probes
96 dir path=opt/SUNWdtrt/tst/common/proc
97 dir path=opt/SUNWdtrt/tst/common/profile-n
98 dir path=opt/SUNWdtrt/tst/common/providers
99 dir path=opt/SUNWdtrt/tst/common/raise
100 dir path=opt/SUNWdtrt/tst/common/rates
101 dir path=opt/SUNWdtrt/tst/common/safety
102 dir path=opt/SUNWdtrt/tst/common/scalars
103 dir path=opt/SUNWdtrt/tst/common/sched
104 dir path=opt/SUNWdtrt/tst/common/scripting
105 dir path=opt/SUNWdtrt/tst/common/sdt
106 dir path=opt/SUNWdtrt/tst/common/sizeof
107 dir path=opt/SUNWdtrt/tst/common/speculation
108 dir path=opt/SUNWdtrt/tst/common/stability
109 dir path=opt/SUNWdtrt/tst/common/stack
110 dir path=opt/SUNWdtrt/tst/common/stackdepth
111 dir path=opt/SUNWdtrt/tst/common/stop
112 dir path=opt/SUNWdtrt/tst/common/strlen
113 dir path=opt/SUNWdtrt/tst/common/strtoll
114 dir path=opt/SUNWdtrt/tst/common/struct
115 dir path=opt/SUNWdtrt/tst/common/sugar
116 dir path=opt/SUNWdtrt/tst/common/syscall
117 dir path=opt/SUNWdtrt/tst/common/sysevent
118 dir path=opt/SUNWdtrt/tst/common/tick-n
119 dir path=opt/SUNWdtrt/tst/common/trace
120 dir path=opt/SUNWdtrt/tst/common/tracemem
```

```

121 dir path=opt/SUNWdtrt/tst/common/translators
122 dir path=opt/SUNWdtrt/tst/common/typedef
123 dir path=opt/SUNWdtrt/tst/common/types
124 dir path=opt/SUNWdtrt/tst/common/uctf
125 dir path=opt/SUNWdtrt/tst/common/union
126 dir path=opt/SUNWdtrt/tst/common/usdt
127 dir path=opt/SUNWdtrt/tst/common/ustack
128 dir path=opt/SUNWdtrt/tst/common/vars
129 dir path=opt/SUNWdtrt/tst/common/version
130 $(i386_ONLY)dir path=opt/SUNWdtrt/tst/i86xpv
131 $(i386_ONLY)dir path=opt/SUNWdtrt/tst/i86xpv/xdt
132 file path=opt/SUNWdtrt/README mode=0444
133 file path=opt/SUNWdtrt/bin/$(ARCH32)/chkargs mode=0555
134 file path=opt/SUNWdtrt/bin/$(ARCH64)/chkargs mode=0555
135 file path=opt/SUNWdtrt/bin/baddof mode=0555
136 file path=opt/SUNWdtrt/bin/badioc1 mode=0555
137 file path=opt/SUNWdtrt/bin/chkargs mode=0555
138 file path=opt/SUNWdtrt/bin/dstyle mode=0555
139 file path=opt/SUNWdtrt/bin/dtest mode=0555
140 file path=opt/SUNWdtrt/bin/dtfailures mode=0555
141 file path=opt/SUNWdtrt/bin/exception.lst mode=0444
142 file path=opt/SUNWdtrt/bin/jdtrace mode=0555
143 file path=opt/SUNWdtrt/lib/java/jdtrace.jar
144 file path=opt/SUNWdtrt/tst/$(ARCH)/arrays/tst.uregsarray.d mode=0444
145 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/error/tst.DTRACEFLT_DIVZERO.d \
146 mode=0444
147 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyin.d mode=0444
148 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyinstr.d \
149 mode=0444
150 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyout.d \
151 mode=0444
152 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyoutstr.d \
153 mode=0444
154 $(sparc_ONLY)file \
155 path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.D_PROC_ALIGN.misaligned.d mode=0444
156 $(sparc_ONLY)file \
157 path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.D_PROC_ALIGN.misaligned.exe \
158 mode=0555
159 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.d mode=0444
160 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.exe mode=0555
161 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d mode=0444
162 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d.out mode=0444
163 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.exe mode=0555
164 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.d mode=0444
165 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.exe mode=0555
166 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.d mode=0444
167 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.exe mode=0555
168 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.d mode=0444
169 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.exe mode=0555
170 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.exe mode=0555
171 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.ksh mode=0444
172 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/usdt/tst.tailcall.ksh \
173 mode=0444
174 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d mode=0444
175 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d.out mode=0444
176 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.exe mode=0555
177 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.d mode=0444
178 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.exe mode=0555
179 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d mode=0444
180 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d.out mode=0444
181 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.exe mode=0555
182 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.trapstat.ksh \
183 mode=0444
184 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_FUNC.bad.d mode=0444
185 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_MDIM.bad.d mode=0444
186 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_NULL.bad.d mode=0444

```

```

187 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_REDEF.redef.d mode=0444
188 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.avgtoofew.d mode=0444
189 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.maxnoarg.d mode=0444
190 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.mintoofew.d mode=0444
191 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.quantizetooofew.d \
192 mode=0444
193 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.stddevtoofew.d \
194 mode=0444
195 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.sumtoofew.d mode=0444
196 file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_AGGARG.bad.d mode=0444
197 file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_PROTO.bad.d mode=0444
198 file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_IDENT.bad.d mode=0444
199 file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_UNDEF.badaggfunc.d mode=0444
200 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badexpr.d mode=0444
201 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badkey3.d mode=0444
202 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.noeffect.d mode=0444
203 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey1.d mode=0444
204 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey2.d mode=0444
205 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey4.d mode=0444
206 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqbad1.d \
207 mode=0444
208 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqshort.d \
209 mode=0444
210 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASEVAL.bad.d mode=0444
211 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMTYPE.lqbad1.d mode=0444
212 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMVAL.bad.d mode=0444
213 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.d mode=0444
214 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.order.d \
215 mode=0444
216 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.d mode=0444
217 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.order.d mode=0444
218 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHSTEP.d mode=0444
219 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MISMATCH.lqbadarg.d \
220 mode=0444
221 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPLARGE.lqtoofew.d \
222 mode=0444
223 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPSMALL.bad.d mode=0444
224 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPTYPE.lqbadinc.d \
225 mode=0444
226 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPVAL.bad.d mode=0444
227 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_AGGARG.bad.d mode=0444
228 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_PROTO.bad.d mode=0444
229 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_SCALAR.bad.d mode=0444
230 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_ARG.lquantizetooofew.d \
231 mode=0444
232 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgnoarg.d mode=0444
233 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgtoomany.d mode=0444
234 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.countoomany.d \
235 mode=0444
236 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizenoarg.d \
237 mode=0444
238 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizetoomany.d \
239 mode=0444
240 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxnoarg.d mode=0444
241 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxtoomany.d mode=0444
242 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.minnoarg.d mode=0444
243 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.mintoomany.d mode=0444
244 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizenoarg.d \
245 mode=0444
246 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizetoomany.d \
247 mode=0444
248 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.stddevnoarg.d mode=0444
249 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.stddevtoomany.d \
250 mode=0444
251 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.sumnoarg.d mode=0444
252 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.sumtoomany.d mode=0444

```

```

253 file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_AGGGARG.bad.d mode=0444
254 file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_PROTO.badmany.d mode=0444
255 file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_PROTO.badnone.d mode=0444
256 file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_SCALAR.bad.d mode=0444
257 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggencoding.d mode=0444
258 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggencoding.d.out mode=0444
259 file path=opt/SUNWdtrt/tst/common/aggs/tst.agghist.d mode=0444
260 file path=opt/SUNWdtrt/tst/common/aggs/tst.agghist.d.out mode=0444
261 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpack.d mode=0444
262 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpack.d.out mode=0444
263 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackbanner.ksh mode=0444
264 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackbanner.ksh.out mode=0444
265 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackzoom.d mode=0444
266 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackzoom.d.out mode=0444
267 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggzoom.d mode=0444
268 file path=opt/SUNWdtrt/tst/common/aggs/tst.aggzoom.d.out mode=0444
269 file path=opt/SUNWdtrt/tst/common/aggs/tst.allquant.d mode=0444
270 file path=opt/SUNWdtrt/tst/common/aggs/tst.allquant.d.out mode=0444
271 file path=opt/SUNWdtrt/tst/common/aggs/tst.avg.d mode=0444
272 file path=opt/SUNWdtrt/tst/common/aggs/tst.avg.d.out mode=0444
273 file path=opt/SUNWdtrt/tst/common/aggs/tst.avg_neg.d mode=0444
274 file path=opt/SUNWdtrt/tst/common/aggs/tst.avg_neg.d.out mode=0444
275 file path=opt/SUNWdtrt/tst/common/aggs/tst.clear.d mode=0444
276 file path=opt/SUNWdtrt/tst/common/aggs/tst.clear.d.out mode=0444
277 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg.d mode=0444
278 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg.d.out mode=0444
279 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg2.d mode=0444
280 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg2.d.out mode=0444
281 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearthenormalize.d mode=0444
282 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearthenormalize.d.out mode=0444
283 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearlquantize.d mode=0444
284 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearlquantize.d.out mode=0444
285 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearnormalize.d mode=0444
286 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearnormalize.d.out mode=0444
287 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearstddev.d mode=0444
288 file path=opt/SUNWdtrt/tst/common/aggs/tst.clearstddev.d.out mode=0444
289 file path=opt/SUNWdtrt/tst/common/aggs/tst.count.d mode=0444
290 file path=opt/SUNWdtrt/tst/common/aggs/tst.count.d.out mode=0444
291 file path=opt/SUNWdtrt/tst/common/aggs/tst.count2.d mode=0444
292 file path=opt/SUNWdtrt/tst/common/aggs/tst.count2.d.out mode=0444
293 file path=opt/SUNWdtrt/tst/common/aggs/tst.count3.d mode=0444
294 file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalize.d mode=0444
295 file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalize.d.out mode=0444
296 file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalizeonly.d mode=0444
297 file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalizeonly.d.out mode=0444
298 file path=opt/SUNWdtrt/tst/common/aggs/tst.fmtnormalize.d mode=0444
299 file path=opt/SUNWdtrt/tst/common/aggs/tst.fmtnormalize.d.out mode=0444
300 file path=opt/SUNWdtrt/tst/common/aggs/tst.forms.d mode=0444
301 file path=opt/SUNWdtrt/tst/common/aggs/tst.forms.d.out mode=0444
302 file path=opt/SUNWdtrt/tst/common/aggs/tst.goodkey.d mode=0444
303 file path=opt/SUNWdtrt/tst/common/aggs/tst.keysort.d mode=0444
304 file path=opt/SUNWdtrt/tst/common/aggs/tst.keysort.d.out mode=0444
305 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantize.d mode=0444
306 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantize.d.out mode=0444
307 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantnormal.d mode=0444
308 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantnormal.d.out mode=0444
309 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantrange.d mode=0444
310 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantrange.d.out mode=0444
311 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantround.d mode=0444
312 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantround.d.out mode=0444
313 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantzero.d mode=0444
314 file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantzero.d.out mode=0444
315 file path=opt/SUNWdtrt/tst/common/aggs/tst.max.d mode=0444
316 file path=opt/SUNWdtrt/tst/common/aggs/tst.max.d.out mode=0444
317 file path=opt/SUNWdtrt/tst/common/aggs/tst.max_neg.d mode=0444
318 file path=opt/SUNWdtrt/tst/common/aggs/tst.max_neg.d.out mode=0444

```

```

319 file path=opt/SUNWdtrt/tst/common/aggs/tst.min.d mode=0444
320 file path=opt/SUNWdtrt/tst/common/aggs/tst.min.d.out mode=0444
321 file path=opt/SUNWdtrt/tst/common/aggs/tst.min_neg.d mode=0444
322 file path=opt/SUNWdtrt/tst/common/aggs/tst.min_neg.d.out mode=0444
323 file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs1.d mode=0444
324 file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs2.d mode=0444
325 file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs2.d.out mode=0444
326 file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs3.d mode=0444
327 file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs3.d.out mode=0444
328 file path=opt/SUNWdtrt/tst/common/aggs/tst.multinormalize.d mode=0444
329 file path=opt/SUNWdtrt/tst/common/aggs/tst.multinormalize.d.out mode=0444
330 file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d mode=0444
331 file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d.out mode=0444
332 file path=opt/SUNWdtrt/tst/common/aggs/tst.negorder.d mode=0444
333 file path=opt/SUNWdtrt/tst/common/aggs/tst.negorder.d.out mode=0444
334 file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d mode=0444
335 file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d.out mode=0444
336 file path=opt/SUNWdtrt/tst/common/aggs/tst.negtrunc.d mode=0444
337 file path=opt/SUNWdtrt/tst/common/aggs/tst.negtrunc.d.out mode=0444
338 file path=opt/SUNWdtrt/tst/common/aggs/tst.negtruncquant.d mode=0444
339 file path=opt/SUNWdtrt/tst/common/aggs/tst.negtruncquant.d.out mode=0444
340 file path=opt/SUNWdtrt/tst/common/aggs/tst.normalize.d mode=0444
341 file path=opt/SUNWdtrt/tst/common/aggs/tst.normalize.d.out mode=0444
342 file path=opt/SUNWdtrt/tst/common/aggs/tst.order.d mode=0444
343 file path=opt/SUNWdtrt/tst/common/aggs/tst.order.d.out mode=0444
344 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantize.d mode=0444
345 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantize.d.out mode=0444
346 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantmany.d mode=0444
347 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantmany.d.out mode=0444
348 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantround.d mode=0444
349 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantround.d.out mode=0444
350 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantzero.d mode=0444
351 file path=opt/SUNWdtrt/tst/common/aggs/tst.quantzero.d.out mode=0444
352 file path=opt/SUNWdtrt/tst/common/aggs/tst.signature.d mode=0444
353 file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeys.d mode=0444
354 file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeys.d.out mode=0444
355 file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeyspos.d mode=0444
356 file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeyspos.d.out mode=0444
357 file path=opt/SUNWdtrt/tst/common/aggs/tst.sizedkeys.d mode=0444
358 file path=opt/SUNWdtrt/tst/common/aggs/tst.sizedkeys.d.out mode=0444
359 file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.d mode=0444
360 file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.d.out mode=0444
361 file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.normalize.d mode=0444
362 file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.normalize.d.out mode=0444
363 file path=opt/SUNWdtrt/tst/common/aggs/tst.subr.d mode=0444
364 file path=opt/SUNWdtrt/tst/common/aggs/tst.sum.d mode=0444
365 file path=opt/SUNWdtrt/tst/common/aggs/tst.sum.d.out mode=0444
366 file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc.d mode=0444
367 file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc.d.out mode=0444
368 file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc0.d mode=0444
369 file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc0.d.out mode=0444
370 file path=opt/SUNWdtrt/tst/common/aggs/tst.truncquant.d mode=0444
371 file path=opt/SUNWdtrt/tst/common/aggs/tst.truncquant.d.out mode=0444
372 file path=opt/SUNWdtrt/tst/common/aggs/tst.valsortkeypos.d mode=0444
373 file path=opt/SUNWdtrt/tst/common/aggs/tst.valsortkeypos.d.out mode=0444
374 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0.d mode=0444
375 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0_1.d \
mode=0444
376
377 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0_2.d \
mode=0444
378
379 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.modby0.d mode=0444
380 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.addmin.d mode=0444
381 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.divmin.d mode=0444
382 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muladd.d mode=0444
383 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muldiv.d mode=0444
384 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d mode=0444

```

```

385 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d.out mode=0444
386 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d mode=0444
387 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d.out mode=0444
388 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compnarrowassign.d mode=0444
389 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compnarrowassign.d.out \
390 mode=0444
391 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d mode=0444
392 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d.out mode=0444
393 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.nonarrow.ksh mode=0444
394 file path=opt/SUNWdtrt/tst/common/arrays/err.D_ARR_BADREF.bad.d mode=0444
395 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRBIG.toobig.d mode=0444
396 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRNULL.bad.d mode=0444
397 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRSUB.bad.d mode=0444
398 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_PROTO_TYPE.badtuple.d \
399 mode=0444
400 file path=opt/SUNWdtrt/tst/common/arrays/err.D_IDENT_UNDEF.badureg.d mode=0444
401 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic1.d mode=0444
402 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic2.d mode=0444
403 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic3.d mode=0444
404 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic4.d mode=0444
405 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic5.d mode=0444
406 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic6.d mode=0444
407 file path=opt/SUNWdtrt/tst/common/arrays/tst.uregsarray.d mode=0444
408 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupgtype.d \
409 mode=0444
410 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupdtype.d \
411 mode=0444
412 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.this.d mode=0444
413 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_ARG.badsig.d mode=0444
414 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toofew.d mode=0444
415 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toomany.d mode=0444
416 file path=opt/SUNWdtrt/tst/common/assocs/err.D_SYNTAX.errassign.d mode=0444
417 file path=opt/SUNWdtrt/tst/common/assocs/err.tupoflow.d mode=0444
418 file path=opt/SUNWdtrt/tst/common/assocs/tst.cpyarray.d mode=0444
419 file path=opt/SUNWdtrt/tst/common/assocs/tst.diffprofile.d mode=0444
420 file path=opt/SUNWdtrt/tst/common/assocs/tst.initialize.d mode=0444
421 file path=opt/SUNWdtrt/tst/common/assocs/tst.invalidref.d mode=0444
422 file path=opt/SUNWdtrt/tst/common/assocs/tst.misc.d mode=0444
423 file path=opt/SUNWdtrt/tst/common/assocs/tst.orthogonality.d mode=0444
424 file path=opt/SUNWdtrt/tst/common/assocs/tst.this.d mode=0444
425 file path=opt/SUNWdtrt/tst/common/assocs/tst.valassign.d.out mode=0444
426 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.begin.d mode=0444
427 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.tick.d mode=0444
428 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d mode=0444
429 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d.out mode=0444
430 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d mode=0444
431 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d.out mode=0444
432 file \
433 path=opt/SUNWdtrt/tst/common/bitfields/err.D_ADDROF_BITFIELD.BitfieldAddress
434 mode=0444
435 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.NegBitField.d \
436 mode=0444
437 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.ZeroBitField.d \
438 mode=0444
439 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.ExceedBaseType.d \
440 mode=0444
441 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.GreaterThan64.d \
442 mode=0444
443 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFTYPE.badtype.d \
444 mode=0444
445 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_OFFSETOF_BITFIELD.d \
446 mode=0444
447 file \
448 path=opt/SUNWdtrt/tst/common/bitfields/err.D_SIZEOF_BITFIELD.SizeofBitfield.
449 mode=0444
450 file path=opt/SUNWdtrt/tst/common/bitfields/tst.BitFieldPromotion.d mode=0444

```

```

451 file path=opt/SUNWdtrt/tst/common/bitfields/tst.SizeofBitField.d mode=0444
452 file path=opt/SUNWdtrt/tst/common/buffering/err.end.d mode=0444
453 file path=opt/SUNWdtrt/tst/common/buffering/err.resize1.d mode=0444
454 file path=opt/SUNWdtrt/tst/common/buffering/err.resize2.d mode=0444
455 file path=opt/SUNWdtrt/tst/common/buffering/err.resize3.d mode=0444
456 file path=opt/SUNWdtrt/tst/common/buffering/err.zerobuf.d mode=0444
457 file path=opt/SUNWdtrt/tst/common/buffering/tst.aligning.d mode=0444
458 file path=opt/SUNWdtrt/tst/common/buffering/tst.cputime.ksh mode=0444
459 file path=opt/SUNWdtrt/tst/common/buffering/tst.dynvarsize.d mode=0444
460 file path=opt/SUNWdtrt/tst/common/buffering/tst.fill1.d mode=0444
461 file path=opt/SUNWdtrt/tst/common/buffering/tst.fill1.d.out mode=0444
462 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize1.d mode=0444
463 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize2.d mode=0444
464 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize3.d mode=0444
465 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring1.d mode=0444
466 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring2.d mode=0444
467 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring2.d.out mode=0444
468 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring3.d mode=0444
469 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring3.d.out mode=0444
470 file path=opt/SUNWdtrt/tst/common/buffering/tst.smallring.d mode=0444
471 file path=opt/SUNWdtrt/tst/common/buffering/tst.switch1.d mode=0444
472 file path=opt/SUNWdtrt/tst/common/buffering/tst.switch1.d.out mode=0444
473 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.cpuusage.d \
474 mode=0444
475 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.nice.d \
476 mode=0444
477 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.priority.d \
478 mode=0444
479 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.prsize.d \
480 mode=0444
481 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.rssize.d \
482 mode=0444
483 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg0.d mode=0444
484 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg0clause.d mode=0444
485 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1.d mode=0444
486 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1to8.d mode=0444
487 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1to8clause.d mode=0444
488 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.caller.d mode=0444
489 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.caller1.d mode=0444
490 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.epid.d mode=0444
491 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.epid1.d mode=0444
492 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.errno.d mode=0444
493 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.errno1.d mode=0444
494 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.execname.d mode=0444
495 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.hpriority.d mode=0444
496 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.id.d mode=0444
497 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.id1.d mode=0444
498 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.ipl.d mode=0444
499 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.ipl1.d mode=0444
500 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.lwpsinfo.d mode=0444
501 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.lwpsinfo1.d mode=0444
502 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.lwpsinfo.d mode=0444
503 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.pid1.d mode=0444
504 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.psinfo.d mode=0444
505 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.psinfo1.d mode=0444
506 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.tid.d mode=0444
507 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.tid1.d mode=0444
508 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.timestamp.d mode=0444
509 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.vtimestamp.d mode=0444
510 file path=opt/SUNWdtrt/tst/common/cg/err.D_NOREG.noreg.d mode=0444
511 file path=opt/SUNWdtrt/tst/common/cg/err.baddif.d mode=0444
512 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.aggfun.d mode=0444
513 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.aggtup.d mode=0444
514 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.arrrup.d mode=0444
515 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.body.d mode=0444
516 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.both.d mode=0444

```



```

517 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.pred.d mode=0444
518 file path=opt/SUNWdtrt/tst/common/clauses/tst.nopred.d mode=0444
519 file path=opt/SUNWdtrt/tst/common/clauses/tst.pred.d mode=0444
520 file path=opt/SUNWdtrt/tst/common/clauses/tst.predfirst.d mode=0444
521 file path=opt/SUNWdtrt/tst/common/clauses/tst.predlast.d mode=0444
522 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.lowfrequency.d \
523 mode=0444
524 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.malformedoverflow.d \
525 mode=0444
526 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.nonexistentevent.d \
527 mode=0444
528 file path=opt/SUNWdtrt/tst/common/cpc/err.cpcvscpustatpart1.ksh mode=0444
529 file path=opt/SUNWdtrt/tst/common/cpc/err.cpcvscpustatpart2.ksh mode=0444
530 file path=opt/SUNWdtrt/tst/common/cpc/err.cputrackfailtostart.ksh mode=0444
531 file path=opt/SUNWdtrt/tst/common/cpc/err.cputrackterminates.ksh mode=0444
532 file path=opt/SUNWdtrt/tst/common/cpc/err.toomanyenablings.d mode=0444
533 file path=opt/SUNWdtrt/tst/common/cpc/tst.allcpus.ksh mode=0444
534 file path=opt/SUNWdtrt/tst/common/cpc/tst.genericevent.d mode=0444
535 file path=opt/SUNWdtrt/tst/common/cpc/tst.platformevent.ksh mode=0444
536 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_LOCASSC.NonLocalAssoc.d \
537 mode=0444
538 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_LONGINT.LongStruct.d \
539 mode=0444
540 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PARMCLASS.BadStorageClass.d \
541 mode=0444
542 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_NAME.VoidName.d \
543 mode=0444
544 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_TYPE.Dyn.d mode=0444
545 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_VARARGS.VarLenArgs.d \
546 mode=0444
547 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_VOID.NonSoleVoid.d \
548 mode=0444
549 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_SIGNINT.UnsignedStruct.d \
550 mode=0444
551 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_VOIDATTR.ShortVoidDecl.d \
552 mode=0444
553 file path=opt/SUNWdtrt/tst/common/decls/tst.arrays.d mode=0444
554 file path=opt/SUNWdtrt/tst/common/decls/tst.basics.d mode=0444
555 file path=opt/SUNWdtrt/tst/common/decls/tst.funcs.d mode=0444
556 file path=opt/SUNWdtrt/tst/common/decls/tst.pointers.d mode=0444
557 file path=opt/SUNWdtrt/tst/common/decls/tst.varargsfuncs.d mode=0444
558 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_AGGREGATION.d mode=0444
559 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_DBLERROR.d mode=0444
560 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_DYNAMIC.d mode=0444
561 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_PRINCIPAL.d mode=0444
562 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_PRINCIPAL.end.d \
563 mode=0444
564 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_SPEC.d mode=0444
565 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_SPECUNAVAIL.d mode=0444
566 file path=opt/SUNWdtrt/tst/common/drops/drpf.DTRACEDROP_STKSTROVERFLOW.d \
567 mode=0444
568 file \
569 path=opt/SUNWdtrt/tst/common/dtraceUtil/err.D_PDESC_ZERO.InvalidDescription1
570 mode=0444
571 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.AddSearchPath.d.ksh mode=0444
572 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeGiga.d.ksh mode=0444
573 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeKilo.d.ksh mode=0444
574 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeMega.d.ksh mode=0444
575 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeTera.d.ksh mode=0444
576 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DataModel32.d.ksh mode=0444
577 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DataModel64.d.ksh mode=0444
578 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh \
579 mode=0444
580 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh.out \
581 mode=0444
582 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh \

```

```

583 mode=0444
584 file \
585 path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh.out \
586 mode=0444
587 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh \
588 mode=0444
589 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh.out \
590 mode=0444
591 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh \
592 mode=0444
593 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh.out \
594 mode=0444
595 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh \
596 mode=0444
597 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh.out \
598 mode=0444
599 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh \
600 mode=0444
601 file \
602 path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh.out \
603 mode=0444
604 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithoutW.d.ksh \
605 mode=0444
606 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationOut.d.ksh \
607 mode=0444
608 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationWithO.d.ksh \
609 mode=0444
610 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus1.d.ksh mode=0444
611 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus2.d.ksh mode=0444
612 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExtraneousProbeIds.d.ksh \
613 mode=0444
614 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName1.d.ksh \
615 mode=0444
616 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName2.d.ksh \
617 mode=0444
618 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId1.d.ksh mode=0444
619 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId2.d.ksh mode=0444
620 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId3.d.ksh mode=0444
621 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule1.d.ksh \
622 mode=0444
623 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule2.d.ksh \
624 mode=0444
625 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule3.d.ksh \
626 mode=0444
627 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule4.d.ksh \
628 mode=0444
629 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProbeIdentifier.d.ksh \
630 mode=0444
631 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider1.d.ksh \
632 mode=0444
633 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider2.d.ksh \
634 mode=0444
635 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider3.d.ksh \
636 mode=0444
637 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider4.d.ksh \
638 mode=0444
639 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc1.d.ksh \
640 mode=0444
641 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc2.d.ksh \
642 mode=0444
643 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc3.d.ksh \
644 mode=0444
645 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc4.d.ksh \
646 mode=0444
647 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc5.d.ksh \
648 mode=0444

```

```

649 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc6.d.ksh \
650 mode=0444
651 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc7.d.ksh \
652 mode=0444
653 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc8.d.ksh \
654 mode=0444
655 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc9.d.ksh \
656 mode=0444
657 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID1.d.ksh \
658 mode=0444
659 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID2.d.ksh \
660 mode=0444
661 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID3.d.ksh \
662 mode=0444
663 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID4.d.ksh \
664 mode=0444
665 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID5.d.ksh \
666 mode=0444
667 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID6.d.ksh \
668 mode=0444
669 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID7.d.ksh \
670 mode=0444
671 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule1.d.ksh \
672 mode=0444
673 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule2.d.ksh \
674 mode=0444
675 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule3.d.ksh \
676 mode=0444
677 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule4.d.ksh \
678 mode=0444
679 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule5.d.ksh \
680 mode=0444
681 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule6.d.ksh \
682 mode=0444
683 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule7.d.ksh \
684 mode=0444
685 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule8.d.ksh \
686 mode=0444
687 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName1.d.ksh \
688 mode=0444
689 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName2.d.ksh \
690 mode=0444
691 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName3.d.ksh \
692 mode=0444
693 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName4.d.ksh \
694 mode=0444
695 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName5.d.ksh \
696 mode=0444
697 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName6.d.ksh \
698 mode=0444
699 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName7.d.ksh \
700 mode=0444
701 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName8.d.ksh \
702 mode=0444
703 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName9.d.ksh \
704 mode=0444
705 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider1.d.ksh \
706 mode=0444
707 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider2.d.ksh \
708 mode=0444
709 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider3.d.ksh \
710 mode=0444
711 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider4.d.ksh \
712 mode=0444
713 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider5.d.ksh \
714 mode=0444

```

```

715 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.MultipleInvalidProbeId.d.ksh \
716 mode=0444
717 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.PreprocessorStatement.d.ksh \
718 mode=0444
719 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh mode=0444
720 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh.out mode=0444
721 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh mode=0444
722 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh.out \
723 mode=0444
724 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.UndefineNameWithCPP.d.ksh \
725 mode=0444
726 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh \
727 mode=0444
728 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh.out \
729 mode=0444
730 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh \
731 mode=0444
732 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh.out \
733 mode=0444
734 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh \
735 mode=0444
736 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh.out \
737 mode=0444
738 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbeIdentifier.d.ksh \
739 mode=0444
740 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbesWithoutZ.d.ksh \
741 mode=0444
742 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh \
743 mode=0444
744 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh.out \
745 mode=0444
746 file path=opt/SUNWdtrt/tst/common/end/err.D_IDENT_UNDEF.timespent.d mode=0444
747 file path=opt/SUNWdtrt/tst/common/end/tst.end.d mode=0444
748 file path=opt/SUNWdtrt/tst/common/end/tst.endwithoutbegin.d mode=0444
749 file path=opt/SUNWdtrt/tst/common/end/tst.multibeginend.d mode=0444
750 file path=opt/SUNWdtrt/tst/common/end/tst.multiend.d mode=0444
751 file path=opt/SUNWdtrt/tst/common/enum/err.D_DECL_IDRED.EnumSameName.d \
752 mode=0444
753 file path=opt/SUNWdtrt/tst/common/enum/err.D_UNKNOWN.RepeatIdentifiers.d \
754 mode=0444
755 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumEquality.d mode=0444
756 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumSameValue.d mode=0444
757 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumValAssign.d mode=0444
758 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.setfromscript.d \
759 mode=0444
760 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.unsetfromscript.d \
761 mode=0444
762 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh mode=0444
763 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh.out mode=0444
764 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh mode=0444
765 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh.out mode=0444
766 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh mode=0444
767 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh.out mode=0444
768 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh mode=0444
769 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh.out mode=0444
770 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh mode=0444
771 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh.out mode=0444
772 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_BADADDR.d mode=0444
773 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_DIVZERO.d mode=0444
774 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_UNKNOWN.d mode=0444
775 file path=opt/SUNWdtrt/tst/common/error/tst.error.d mode=0444
776 file path=opt/SUNWdtrt/tst/common/error/tst.errrend.d mode=0444
777 file path=opt/SUNWdtrt/tst/common/exit/err.D_PROTO_LEN.noarg.d mode=0444
778 file path=opt/SUNWdtrt/tst/common/exit/err.exitarg1.d mode=0444
779 file path=opt/SUNWdtrt/tst/common/exit/tst.basic1.d mode=0444
780 file path=opt/SUNWdtrt/tst/common/fbtprovider/err.D_PDESC_ZERO.notreturn.d \

```

```

781 mode=0444
782 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.basic.d mode=0444
783 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionentry.d mode=0444
784 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionreturnvalue.d \
785 mode=0444
786 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.ioctlargs.d mode=0444
787 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offset.d mode=0444
788 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offsetzero.d mode=0444
789 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return.d mode=0444
790 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return0.d mode=0444
791 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.tailcall.d mode=0444
792 file path=opt/SUNWdtrt/tst/common/funcs/err.D_FUNC_UNDEF.progenyofbad1.d \
793 mode=0444
794 file path=opt/SUNWdtrt/tst/common/funcs/err.D_OP_VFPTR.badop.d mode=0444
795 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.chillbadarg.d \
796 mode=0444
797 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.copyoutbadarg.d \
798 mode=0444
799 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.mobadarg.d mode=0444
800 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.raisebadarg.d \
801 mode=0444
802 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.tolower.d mode=0444
803 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.toupper.d mode=0444
804 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.allocanoarg.d \
805 mode=0444
806 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.badbreakpoint.d \
807 mode=0444
808 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoofew.d \
809 mode=0444
810 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoomany.d \
811 mode=0444
812 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrbadarg.d \
813 mode=0444
814 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrtoofew.d \
815 mode=0444
816 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoofew.d \
817 mode=0444
818 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoomany.d \
819 mode=0444
820 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoofew.d mode=0444
821 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoomany.d mode=0444
822 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtabadarg.d mode=0444
823 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoofew.d mode=0444
824 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoomany.d mode=0444
825 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.panicbadarg.d \
826 mode=0444
827 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.progenyofbad2.d \
828 mode=0444
829 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.stopbadarg.d mode=0444
830 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolower.d mode=0444
831 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolowertoomany.d \
832 mode=0444
833 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.toupper.d mode=0444
834 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.touppertoomany.d \
835 mode=0444
836 file path=opt/SUNWdtrt/tst/common/funcs/err.D_STRINGOF_TYPE.badstringof.d \
837 mode=0444
838 file path=opt/SUNWdtrt/tst/common/funcs/err.D_VAR_UNDEF.badvar.d mode=0444
839 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca.d mode=0444
840 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca2.d mode=0444
841 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy.d mode=0444
842 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy1.d mode=0444
843 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy2.d mode=0444
844 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy3.d mode=0444
845 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy4.d mode=0444
846 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy5.d mode=0444

```

```

847 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy6.d mode=0444
848 file path=opt/SUNWdtrt/tst/common/funcs/err.badchill.d mode=0444
849 file path=opt/SUNWdtrt/tst/common/funcs/err.chillbadarg.ksh mode=0444
850 file path=opt/SUNWdtrt/tst/common/funcs/err.copypout.d mode=0444
851 file path=opt/SUNWdtrt/tst/common/funcs/err.copypoutbadaddr.ksh mode=0444
852 file path=opt/SUNWdtrt/tst/common/funcs/err.copypoutstrbadaddr.ksh mode=0444
853 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoa6badaddr.d mode=0444
854 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoa6badaddr.d mode=0444
855 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadaddr.d mode=0444
856 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadarg.d mode=0444
857 file path=opt/SUNWdtrt/tst/common/funcs/tst.badfireopen.ksh mode=0444
858 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d mode=0444
859 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d.out mode=0444
860 file path=opt/SUNWdtrt/tst/common/funcs/tst.bcopy.d mode=0444
861 file path=opt/SUNWdtrt/tst/common/funcs/tst.chill.ksh mode=0444
862 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d mode=0444
863 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d.out mode=0444
864 file path=opt/SUNWdtrt/tst/common/funcs/tst.copypin.d mode=0444
865 file path=opt/SUNWdtrt/tst/common/funcs/tst.copypinto.d mode=0444
866 file path=opt/SUNWdtrt/tst/common/funcs/tst.ddi_pathname.d mode=0444
867 file path=opt/SUNWdtrt/tst/common/funcs/tst.default.d mode=0444
868 file path=opt/SUNWdtrt/tst/common/funcs/tst.freopen.ksh mode=0444
869 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh mode=0444
870 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh.out mode=0444
871 file path=opt/SUNWdtrt/tst/common/funcs/tst.hton.d mode=0444
872 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d mode=0444
873 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d.out mode=0444
874 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d mode=0444
875 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d.out mode=0444
876 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d mode=0444
877 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d.out mode=0444
878 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d mode=0444
879 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d.out mode=0444
880 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d mode=0444
881 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d.out mode=0444
882 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d mode=0444
883 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d.out mode=0444
884 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owned.d mode=0444
885 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owner.d mode=0444
886 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_type_adaptive.d mode=0444
887 file path=opt/SUNWdtrt/tst/common/funcs/tst.progenyof.d mode=0444
888 file path=opt/SUNWdtrt/tst/common/funcs/tst.rand.d mode=0444
889 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d mode=0444
890 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d.out mode=0444
891 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d mode=0444
892 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d.out mode=0444
893 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d mode=0444
894 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d.out mode=0444
895 file path=opt/SUNWdtrt/tst/common/funcs/tst.startok.d mode=0444
896 file path=opt/SUNWdtrt/tst/common/funcs/tst.startok.d.out mode=0444
897 file path=opt/SUNWdtrt/tst/common/funcs/tst.startok_null.d mode=0444
898 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d mode=0444
899 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d.out mode=0444
900 file path=opt/SUNWdtrt/tst/common/funcs/tst.substminate.d mode=0444
901 file path=opt/SUNWdtrt/tst/common/funcs/tst.substminate.d.out mode=0444
902 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d mode=0444
903 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d.out mode=0444
904 file path=opt/SUNWdtrt/tst/common/funcs/tst.tolower.d mode=0444
905 file path=opt/SUNWdtrt/tst/common/funcs/tst.toupper.d mode=0444
906 file path=opt/SUNWdtrt/tst/common/grammar/err.D_ADDR_DROP_LVAL.d mode=0444
907 file path=opt/SUNWdtrt/tst/common/grammar/err.D_EMPTY.empty.d mode=0444
908 file path=opt/SUNWdtrt/tst/common/grammar/tst.clauses.d mode=0444
909 file path=opt/SUNWdtrt/tst/common/grammar/tst.stmts.d mode=0444
910 file path=opt/SUNWdtrt/tst/common/include/tst.includefirst.ksh mode=0444
911 file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef1.d mode=0444
912 file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef2.d mode=0444

```

```

913 file path=opt/SUNWdtrt/tst/common/inline/err.D_IDENT_UNDEF.recur.d mode=0444
914 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef1.d mode=0444
915 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef2.d mode=0444
916 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.badxlate.d \
917 mode=0444
918 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineDataAssign.d mode=0444
919 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineExpression.d mode=0444
920 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d mode=0444
921 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d.out mode=0444
922 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineTypedef.d mode=0444
923 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineWritableAssign.d mode=0444
924 file path=opt/SUNWdtrt/tst/common/io/tst.fds.d mode=0444
925 file path=opt/SUNWdtrt/tst/common/io/tst.fds.d.out mode=0444
926 file path=opt/SUNWdtrt/tst/common/io/tst.fds.exe mode=0555
927 file path=opt/SUNWdtrt/tst/common/ip/get.ipv4remote.pl mode=0555
928 file path=opt/SUNWdtrt/tst/common/ip/get.ipv6remote.pl mode=0555
929 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh mode=0444
930 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh.out mode=0444
931 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh mode=0444
932 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh.out mode=0444
933 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh mode=0444
934 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh.out mode=0444
935 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh mode=0444
936 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh.out mode=0444
937 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh mode=0444
938 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh.out mode=0444
939 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh mode=0444
940 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh.out mode=0444
941 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh mode=0444
942 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh.out mode=0444
943 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remotetcp.ksh mode=0444
944 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remotetcp.ksh.out mode=0444
945 file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh mode=0444
946 file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh.out mode=0444
947 file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh mode=0444
948 file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh.out mode=0444
949 file path=opt/SUNWdtrt/tst/common/java_api/test.jar
950 file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh mode=0444
951 file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh.out mode=0444
952 file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh mode=0444
953 file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh.out mode=0444
954 file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh mode=0444
955 file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh.out mode=0444
956 file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh mode=0444
957 file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh.out mode=0444
958 file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh mode=0444
959 file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh.out mode=0444
960 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.exe mode=0555
961 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh mode=0444
962 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh.out \
963 mode=0444
964 file path=opt/SUNWdtrt/tst/common/java_api/tst.GetAggregate.ksh mode=0444
965 file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh mode=0444
966 file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh.out mode=0444
967 file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh mode=0444
968 file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh.out \
969 mode=0444
970 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.exe mode=0555
971 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh mode=0444
972 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh.out mode=0444
973 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh mode=0444
974 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh.out \
975 mode=0444
976 file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh mode=0444
977 file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh.out mode=0444
978 file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh mode=0444

```

```

979 file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh.out mode=0444
980 file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d mode=0444
981 file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d.out mode=0444
982 file path=opt/SUNWdtrt/tst/common/json/tst.general.d mode=0444
983 file path=opt/SUNWdtrt/tst/common/json/tst.general.d.out mode=0444
984 file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d mode=0444
985 file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d.out mode=0444
986 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d mode=0444
987 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d.out mode=0444
988 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.exe mode=0555
989 file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NL.char.d mode=0444
990 file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NULL.char.d mode=0444
991 file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_DIGIT.InvalidDigit.d \
992 mode=0444
993 file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_OFLOW.BigInt.d mode=0444
994 file path=opt/SUNWdtrt/tst/common/lexer/err.D_STR_NL.string.d mode=0444
995 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack1.d mode=0444
996 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack2.d mode=0444
997 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack1.d mode=0444
998 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack2.d mode=0444
999 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack3.d mode=0444
1000 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren1.d mode=0444
1001 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren2.d mode=0444
1002 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren3.d mode=0444
1003 file path=opt/SUNWdtrt/tst/common/lexer/tst.D_MACRO_OFLOW.ParIntOvflow.d.ksh \
1004 mode=0444
1005 file \
1006 path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR EVEN.nodivide.d
1007 mode=0444
1008 file \
1009 path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR EVEN.notfactor.d
1010 mode=0444
1011 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR MATCH.d \
1012 mode=0444
1013 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR N STEPS.d \
1014 mode=0444
1015 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR SMALL.d \
1016 mode=0444
1017 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR TYPE.d \
1018 mode=0444
1019 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR VAL.d \
1020 mode=0444
1021 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH MATCH.d \
1022 mode=0444
1023 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH TYPE.d \
1024 mode=0444
1025 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH VAL.d mode=0444
1026 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW MATCH.d \
1027 mode=0444
1028 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW TYPE.d mode=0444
1029 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW VAL.d mode=0444
1030 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAGRANGE.d \
1031 mode=0444
1032 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAGTOOBIG.d \
1033 mode=0444
1034 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP MATCH.d \
1035 mode=0444
1036 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP TYPE.d \
1037 mode=0444
1038 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP VAL.d \
1039 mode=0444
1040 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d mode=0444
1041 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d.out mode=0444
1042 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d mode=0444
1043 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d.out mode=0444
1044 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d mode=0444

```

```

1045 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d.out mode=0444
1046 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d mode=0444
1047 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d.out mode=0444
1048 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d mode=0444
1049 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d.out mode=0444
1050 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d mode=0444
1051 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d.out mode=0444
1052 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d mode=0444
1053 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d.out mode=0444
1054 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d mode=0444
1055 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d.out mode=0444
1056 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d mode=0444
1057 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d.out mode=0444
1058 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d mode=0444
1059 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d.out mode=0444
1060 file path=opt/SUNWdtrt/tst/common/mdb/tst.dtracedcmd.ksh mode=0444
1061 file path=opt/SUNWdtrt/tst/common/mib/tst.icmp.ksh mode=0444
1062 file path=opt/SUNWdtrt/tst/common/mib/tst.tcp.ksh mode=0444
1063 file path=opt/SUNWdtrt/tst/common/mib/tst.udp.ksh mode=0444
1064 file path=opt/SUNWdtrt/tst/common/misc/err.D_PRAGMA_OPTSET.d mode=0444
1065 file path=opt/SUNWdtrt/tst/common/misc/tst.badopt.d mode=0444
1066 file path=opt/SUNWdtrt/tst/common/misc/tst.booloop.d mode=0444
1067 file path=opt/SUNWdtrt/tst/common/misc/tst.booloop.d.out mode=0444
1068 file path=opt/SUNWdtrt/tst/common/misc/tst.boofmax.ksh mode=0444
1069 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d mode=0444
1070 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d.out mode=0444
1071 file path=opt/SUNWdtrt/tst/common/misc/tst.enablerace.ksh mode=0444
1072 file path=opt/SUNWdtrt/tst/common/misc/tst.haslam.d mode=0444
1073 file path=opt/SUNWdtrt/tst/common/misc/tst.include.ksh mode=0444
1074 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh mode=0444
1075 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh.out mode=0444
1076 file path=opt/SUNWdtrt/tst/common/misc/tst.roch.d mode=0444
1077 file path=opt/SUNWdtrt/tst/common/misc/tst.schrock.ksh mode=0444
1078 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGKEY.d mode=0444
1079 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGPROTO.d mode=0444
1080 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d mode=0444
1081 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d.out mode=0444
1082 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d mode=0444
1083 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d.out mode=0444
1084 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d mode=0444
1085 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d.out mode=0444
1086 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d mode=0444
1087 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d.out mode=0444
1088 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d mode=0444
1089 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d.out mode=0444
1090 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d mode=0444
1091 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d.out mode=0444
1092 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d mode=0444
1093 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d.out mode=0444
1094 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d mode=0444
1095 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d.out mode=0444
1096 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.d mode=0444
1097 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.exe mode=0555
1098 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.d mode=0444
1099 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.exe mode=0555
1100 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_BITFIELD.bitfield.d \
1101 mode=0444
1102 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.badtype.d \
1103 mode=0444
1104 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.notsou.d \
1105 mode=0444
1106 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.OffsetofNULL.d \
1107 mode=0444
1108 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.badmemb.d mode=0444
1109 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofAlias.d mode=0444
1110 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofArith.d mode=0444

```

```

1111 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofUnion.d mode=0444
1112 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d mode=0444
1113 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d.out mode=0444
1114 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d mode=0444
1115 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d.out mode=0444
1116 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d mode=0444
1117 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d.out mode=0444
1118 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.d mode=0444
1119 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.exe mode=0555
1120 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badprocl.d mode=0444
1121 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_BADPID.badproc2.d mode=0444
1122 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.d mode=0444
1123 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.exe mode=0555
1124 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.d mode=0444
1125 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.exe mode=0555
1126 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.d mode=0444
1127 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.exe mode=0555
1128 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.d mode=0444
1129 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.exe mode=0555
1130 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.d mode=0444
1131 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.exe mode=0555
1132 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.d mode=0444
1133 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.exe mode=0555
1134 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.d mode=0444
1135 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.exe mode=0555
1136 file path=opt/SUNWdtrt/tst/common/pid/tst.addprobes.ksh mode=0444
1137 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.d mode=0444
1138 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.exe mode=0555
1139 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.d mode=0444
1140 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.exe mode=0555
1141 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d mode=0444
1142 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d.out mode=0444
1143 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.exe mode=0555
1144 file path=opt/SUNWdtrt/tst/common/pid/tst.float.d mode=0444
1145 file path=opt/SUNWdtrt/tst/common/pid/tst.float.exe mode=0555
1146 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.d mode=0444
1147 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.exe mode=0555
1148 file path=opt/SUNWdtrt/tst/common/pid/tst.gcc.d mode=0444
1149 file path=opt/SUNWdtrt/tst/common/pid/tst.gcc.exe mode=0555
1148 file path=opt/SUNWdtrt/tst/common/pid/tst.killonerror.ksh mode=0444
1149 file path=opt/SUNWdtrt/tst/common/pid/tst.main.ksh mode=0444
1150 file path=opt/SUNWdtrt/tst/common/pid/tst.manypids.ksh mode=0444
1151 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh mode=0444
1152 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh.out mode=0444
1153 file path=opt/SUNWdtrt/tst/common/pid/tst.probemod.ksh mode=0444
1154 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex1.ksh mode=0444
1155 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh mode=0444
1156 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh.out mode=0444
1157 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh mode=0444
1158 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh.out mode=0444
1159 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh mode=0444
1160 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh.out mode=0444
1161 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.d mode=0444
1162 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.exe mode=0555
1163 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.d mode=0444
1164 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.exe mode=0555
1165 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh mode=0444
1166 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh.out mode=0444
1167 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probemod.ksh mode=0444
1168 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probemod.ksh.out mode=0444
1169 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.d mode=0444
1170 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.exe mode=0555
1171 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.d mode=0444
1172 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.exe mode=0555
1173 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.d mode=0444
1174 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.exe mode=0555

```

```

1175 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.d mode=0444
1176 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.exe mode=0555
1177 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.d mode=0444
1178 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.exe mode=0555
1179 file path=opt/SUNWdtrt/tst/common/pointers/err.BadAlign.d mode=0444
1180 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.ArrayVar.d \
1181 mode=0444
1182 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.DynamicVar.d \
1183 mode=0444
1184 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.agg.d mode=0444
1185 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_NONPTR.noptr.d \
1186 mode=0444
1187 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_VOID.VoidPointerDeref.d \
1188 mode=0444
1189 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_ARRFUN.ArrayAssignment.d \
1190 mode=0444
1191 file \
1192 path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_INCOMPAT.VoidPointerArith.d \
1193 mode=0444
1194 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_LVAL.AddressChange.d \
1195 mode=0444
1196 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.NonPointerAccess.d \
1197 mode=0444
1198 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.badpointer.d mode=0444
1199 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.BadPointerAccess.d \
1200 mode=0444
1201 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.badpointer.d mode=0444
1202 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress1.d mode=0444
1203 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress2.d mode=0444
1204 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress3.d mode=0444
1205 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress4.d mode=0444
1206 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress5.d mode=0444
1207 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer1.d mode=0444
1208 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer2.d mode=0444
1209 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer3.d mode=0444
1210 file path=opt/SUNWdtrt/tst/common/pointers/tst.GlobalVar.d mode=0444
1211 file path=opt/SUNWdtrt/tst/common/pointers/tst.IntegerArithmetic1.d mode=0444
1212 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic1.d mode=0444
1213 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic2.d mode=0444
1214 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic3.d mode=0444
1215 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerAssignment.d mode=0444
1216 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer1.d mode=0444
1217 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer2.d mode=0444
1218 file path=opt/SUNWdtrt/tst/common/pointers/tst.VoidCast.d mode=0444
1219 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast1.d mode=0444
1220 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast2.d mode=0444
1221 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic1.d mode=0444
1222 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic2.d mode=0444
1223 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGERR.d mode=0444
1224 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_DEPEND.main.d mode=0444
1225 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_INVALID.d mode=0444
1226 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_MALFORM.d mode=0444
1227 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_UNUSED.UnusedPragma.d \
1228 mode=0444
1229 file path=opt/SUNWdtrt/tst/common/pragma/err.circlibdep.ksh mode=0444
1230 file path=opt/SUNWdtrt/tst/common/pragma/err.invalidlibdep.ksh mode=0444
1231 file path=opt/SUNWdtrt/tst/common/pragma/tst.libchain.ksh mode=0444
1232 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdep.ksh mode=0444
1233 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepfullyconnected.ksh \
1234 mode=0444
1235 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepsemdir.ksh mode=0444
1236 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal.ksh mode=0444
1237 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal2.ksh mode=0444
1238 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal3.d mode=0444
1239 file path=opt/SUNWdtrt/tst/common/predicates/err.D_PRED_SCALAR.NonScalarPred.d \
1240 mode=0444

```

```

1241 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.invalid.d mode=0444
1242 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.operr.d mode=0444
1243 file path=opt/SUNWdtrt/tst/common/predicates/tst.argsnotcached.d mode=0444
1244 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d mode=0444
1245 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d.out mode=0444
1246 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d mode=0444
1247 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d.out mode=0444
1248 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_IDENT_UNDEF.afterprobe.d \
1249 mode=0444
1250 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_PRAGCTL_INVALID.tabdefine.d \
1251 mode=0444
1252 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_SYNTAX.withoutpound.d \
1253 mode=0444
1254 file path=opt/SUNWdtrt/tst/common/preprocessor/err.defincomp.d mode=0444
1255 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefelsenotendif.d \
1256 mode=0444
1257 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefincomp.d mode=0444
1258 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefnotendif.d mode=0444
1259 file path=opt/SUNWdtrt/tst/common/preprocessor/err.incompelse.d mode=0444
1260 file path=opt/SUNWdtrt/tst/common/preprocessor/err.mulelse.d mode=0444
1261 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d mode=0444
1262 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d.out mode=0444
1263 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d mode=0444
1264 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d.out mode=0444
1265 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d mode=0444
1266 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d.out mode=0444
1267 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d mode=0444
1268 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d.out mode=0444
1269 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d mode=0444
1270 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d.out \
1271 mode=0444
1272 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d mode=0444
1273 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d.out mode=0444
1274 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d mode=0444
1275 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d.out mode=0444
1276 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d mode=0444
1277 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d.out mode=0444
1278 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d mode=0444
1279 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d.out mode=0444
1280 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.predicatedeclare.d \
1281 mode=0444
1282 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d mode=0444
1283 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d.out mode=0444
1284 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d mode=0444
1285 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d.out mode=0444
1286 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d mode=0444
1287 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d.out mode=0444
1288 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d mode=0444
1289 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d.out \
1290 mode=0444
1291 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.withinprobe.d mode=0444
1292 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_AGG.bad.d mode=0444
1293 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_VOID.bad.d mode=0444
1294 file path=opt/SUNWdtrt/tst/common/print/err.D_PROTO_LFN.bad.d mode=0444
1295 file path=opt/SUNWdtrt/tst/common/print/tst.array.d mode=0444
1296 file path=opt/SUNWdtrt/tst/common/print/tst.array.d.out mode=0444
1297 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d mode=0444
1298 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d.out mode=0444
1299 file path=opt/SUNWdtrt/tst/common/print/tst.dyn.d mode=0444
1300 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d mode=0444
1301 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d.out mode=0444
1302 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d mode=0444
1303 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d.out mode=0444
1304 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d mode=0444
1305 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d.out mode=0444
1306 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d mode=0444

```

```

1307 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d.out mode=0444
1308 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badagg.d \
1309 mode=0444
1310 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badfmt.d \
1311 mode=0444
1312 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badval.d \
1313 mode=0444
1314 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_PROTO.bad.d mode=0444
1315 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.jstack.d \
1316 mode=0444
1317 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.stack.d \
1318 mode=0444
1319 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.ustack.d \
1320 mode=0444
1321 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d mode=0444
1322 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d.out mode=0444
1323 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d mode=0444
1324 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d.out mode=0444
1325 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d mode=0444
1326 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d.out mode=0444
1327 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d mode=0444
1328 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d.out mode=0444
1329 file path=opt/SUNWdtrt/tst/common/printa/tst.largeusersym.ksh mode=0444
1330 file path=opt/SUNWdtrt/tst/common/printa/tst.many.d mode=0444
1331 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d mode=0444
1332 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d.out mode=0444
1333 file path=opt/SUNWdtrt/tst/common/printa/tst.stack.d mode=0444
1334 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d mode=0444
1335 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d.out mode=0444
1336 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh mode=0444
1337 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh.out mode=0444
1338 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_AGG_CONV.aggfmt.d \
1339 mode=0444
1340 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.toomany.d \
1341 mode=0444
1342 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.widths.d \
1343 mode=0444
1344 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_FMT.badfmt.d \
1345 mode=0444
1346 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_PROTO.novalue.d \
1347 mode=0444
1348 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.aggarg.d \
1349 mode=0444
1350 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.recursive.d \
1351 mode=0444
1352 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.noprec.d \
1353 mode=0444
1354 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.nowidth.d \
1355 mode=0444
1356 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badprec.d \
1357 mode=0444
1358 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badwidth.d \
1359 mode=0444
1360 file path=opt/SUNWdtrt/tst/common/printf/err.D_PROTO_LEN.toofew.d mode=0444
1361 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv1.d mode=0444
1362 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv2.d mode=0444
1363 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv3.d mode=0444
1364 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d mode=0444
1365 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d.out mode=0444
1366 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d mode=0444
1367 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d.out mode=0444
1368 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d mode=0444
1369 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d.out mode=0444
1370 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d mode=0444
1371 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d.out mode=0444
1372 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d mode=0444

```

```

1373 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d.out mode=0444
1374 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d mode=0444
1375 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d.out mode=0444
1376 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh mode=0444
1377 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh.out mode=0444
1378 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh mode=0444
1379 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh.out mode=0444
1380 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d mode=0444
1381 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d.out mode=0444
1382 file path=opt/SUNWdtrt/tst/common/printf/tst.printE.d mode=0444
1383 file path=opt/SUNWdtrt/tst/common/printf/tst.printE.d.out mode=0444
1384 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d mode=0444
1385 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d.out mode=0444
1386 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d mode=0444
1387 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d.out mode=0444
1388 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d mode=0444
1389 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d.out mode=0444
1390 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d mode=0444
1391 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d.out mode=0444
1392 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d mode=0444
1393 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d.out mode=0444
1394 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d mode=0444
1395 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d.out mode=0444
1396 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d mode=0444
1397 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d.out mode=0444
1398 file path=opt/SUNWdtrt/tst/common/printf/tst.widths1.d mode=0444
1399 file path=opt/SUNWdtrt/tst/common/printf/tst.wp.d mode=0444
1400 file path=opt/SUNWdtrt/tst/common/printf/tst.wp.d.out mode=0444
1401 file path=opt/SUNWdtrt/tst/common/privs/tst.fds.ksh mode=0444
1402 file path=opt/SUNWdtrt/tst/common/privs/tst.func_access.ksh mode=0444
1403 file path=opt/SUNWdtrt/tst/common/privs/tst.getf.ksh mode=0444
1404 file path=opt/SUNWdtrt/tst/common/privs/tst.kpriv.ksh mode=0444
1405 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivdrop.ksh mode=0444
1406 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivrestrict.ksh mode=0444
1407 file path=opt/SUNWdtrt/tst/common/privs/tst.op_access.ksh mode=0444
1408 file path=opt/SUNWdtrt/tst/common/privs/tst.procriv.ksh mode=0444
1409 file path=opt/SUNWdtrt/tst/common/privs/tst.providers.ksh mode=0444
1410 file path=opt/SUNWdtrt/tst/common/privs/tst.tick.ksh mode=0444
1411 file path=opt/SUNWdtrt/tst/common/privs/tst.unpriv_funcs.ksh mode=0444
1412 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probeqtn.d mode=0444
1413 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probestar.d \
1414 mode=0444
1415 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.tickstar.d mode=0444
1416 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.assign.d mode=0444
1417 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declare.d mode=0444
1418 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declarein.d mode=0444
1419 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.lbraces.d mode=0444
1420 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.probespec.d mode=0444
1421 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.rbraces.d mode=0444
1422 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.rdecdec.d mode=0444
1423 file path=opt/SUNWdtrt/tst/common/probes/tst.basic1.d mode=0444
1424 file path=opt/SUNWdtrt/tst/common/probes/tst.check.d mode=0444
1425 file path=opt/SUNWdtrt/tst/common/probes/tst.declare.d mode=0444
1426 file path=opt/SUNWdtrt/tst/common/probes/tst.declareafter.d mode=0444
1427 file path=opt/SUNWdtrt/tst/common/probes/tst.emptyprobe.d mode=0444
1428 file path=opt/SUNWdtrt/tst/common/probes/tst.pragma.d mode=0444
1429 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaaftertab.d mode=0444
1430 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmainside.d mode=0444
1431 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaoutside.d mode=0444
1432 file path=opt/SUNWdtrt/tst/common/probes/tst.probestar.d mode=0444
1433 file path=opt/SUNWdtrt/tst/common/proc/tst.create.ksh mode=0444
1434 file path=opt/SUNWdtrt/tst/common/proc/tst.discard.ksh mode=0444
1435 file path=opt/SUNWdtrt/tst/common/proc/tst.exec.ksh mode=0444
1436 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ENOENT.ksh mode=0444
1437 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ksh mode=0444
1438 file path=opt/SUNWdtrt/tst/common/proc/tst.exitcore.ksh mode=0444

```

```

1439 file path=opt/SUNWdtrt/tst/common/proc/tst.exitexit.ksh mode=0444
1440 file path=opt/SUNWdtrt/tst/common/proc/tst.exitkilled.ksh mode=0444
1441 file path=opt/SUNWdtrt/tst/common/proc/tst.signal.ksh mode=0444
1442 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.d mode=0444
1443 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.exe mode=0555
1444 file path=opt/SUNWdtrt/tst/common/proc/tst.startexit.ksh mode=0444
1445 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZERO.profile.d \
1446 mode=0444
1447 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonens.d mode=0444
1448 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonensec.d \
1449 mode=0444
1450 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROoneus.d mode=0444
1451 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROoneusec.d \
1452 mode=0444
1453 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d mode=0444
1454 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d.out mode=0444
1455 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d mode=0444
1456 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d.out mode=0444
1457 file path=opt/SUNWdtrt/tst/common/profile-n/tst.func.ksh mode=0444
1458 file path=opt/SUNWdtrt/tst/common/profile-n/tst.mod.ksh mode=0444
1459 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilehz.d mode=0444
1460 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilehz.d.out mode=0444
1461 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d mode=0444
1462 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d.out mode=0444
1463 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilemsec.d mode=0444
1464 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilemsec.d.out mode=0444
1465 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenz.d mode=0444
1466 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenz.d.out mode=0444
1467 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d mode=0444
1468 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d.out mode=0444
1469 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1470 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1471 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d mode=0444
1472 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d.out mode=0444
1473 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilesec.d mode=0444
1474 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilesec.d.out mode=0444
1475 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d mode=0444
1476 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d.out mode=0444
1477 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileusec.d mode=0444
1478 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileusec.d.out mode=0444
1479 file path=opt/SUNWdtrt/tst/common/profile-n/tst.sym.ksh mode=0444
1480 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufunc.ksh mode=0444
1481 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.exe mode=0555
1482 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh mode=0444
1483 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh.out mode=0444
1484 file path=opt/SUNWdtrt/tst/common/profile-n/tst.umod.ksh mode=0444
1485 file path=opt/SUNWdtrt/tst/common/profile-n/tst.usym.ksh mode=0444
1486 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_INVALID.wrongdec4.d \
1487 mode=0444
1488 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.nonprofile.d \
1489 mode=0444
1490 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec1.d \
1491 mode=0444
1492 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec2.d \
1493 mode=0444
1494 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec3.d \
1495 mode=0444
1496 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d mode=0444
1497 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d.out mode=0444
1498 file path=opt/SUNWdtrt/tst/common/providers/tst.beginexit.d mode=0444
1499 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d mode=0444
1500 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d.out mode=0444
1501 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d mode=0444
1502 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d.out mode=0444
1503 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d mode=0444
1504 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d.out mode=0444

```

```

1505 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d mode=0444
1506 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d.out mode=0444
1507 file path=opt/SUNWdtrt/tst/common/providers/tst.probename.d mode=0444
1508 file path=opt/SUNWdtrt/tst/common/providers/tst.probename.d.out mode=0444
1509 file path=opt/SUNWdtrt/tst/common/providers/tst.probprov.d mode=0444
1510 file path=opt/SUNWdtrt/tst/common/providers/tst.probprov.d.out mode=0444
1511 file path=opt/SUNWdtrt/tst/common/providers/tst.profund.d mode=0444
1512 file path=opt/SUNWdtrt/tst/common/providers/tst.profund.d.out mode=0444
1513 file path=opt/SUNWdtrt/tst/common/providers/tst.profixit.d mode=0444
1514 file path=opt/SUNWdtrt/tst/common/providers/tst.profixit.d.out mode=0444
1515 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d mode=0444
1516 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d.out mode=0444
1517 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d mode=0444
1518 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d.out mode=0444
1519 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.d mode=0444
1520 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.exe mode=0555
1521 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.d mode=0444
1522 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.exe mode=0555
1523 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.d mode=0444
1524 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.exe mode=0555
1525 file path=opt/SUNWdtrt/tst/common/rates/tst.aggregate.d mode=0444
1526 file path=opt/SUNWdtrt/tst/common/rates/tst.aggregate.d.out mode=0444
1527 file path=opt/SUNWdtrt/tst/common/rates/tst.statusrate.d mode=0444
1528 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d mode=0444
1529 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d.out mode=0444
1530 file path=opt/SUNWdtrt/tst/common/safety/tst.basename.d mode=0444
1531 file path=opt/SUNWdtrt/tst/common/safety/tst.caller.d mode=0444
1532 file path=opt/SUNWdtrt/tst/common/safety/tst.cleanpath.d mode=0444
1533 file path=opt/SUNWdtrt/tst/common/safety/tst.copyin.d mode=0444
1534 file path=opt/SUNWdtrt/tst/common/safety/tst.copyin2.d mode=0444
1535 file path=opt/SUNWdtrt/tst/common/safety/tst.ddi_pathname.d mode=0444
1536 file path=opt/SUNWdtrt/tst/common/safety/tst.dirname.d mode=0444
1537 file path=opt/SUNWdtrt/tst/common/safety/tst.errno.d mode=0444
1538 file path=opt/SUNWdtrt/tst/common/safety/tst.execname.d mode=0444
1539 file path=opt/SUNWdtrt/tst/common/safety/tst.gitid.d mode=0444
1540 file path=opt/SUNWdtrt/tst/common/safety/tst.hton.d mode=0444
1541 file path=opt/SUNWdtrt/tst/common/safety/tst.index.d mode=0444
1542 file path=opt/SUNWdtrt/tst/common/safety/tst.msgdsize.d mode=0444
1543 file path=opt/SUNWdtrt/tst/common/safety/tst.mgszsize.d mode=0444
1544 file path=opt/SUNWdtrt/tst/common/safety/tst.null.d mode=0444
1545 file path=opt/SUNWdtrt/tst/common/safety/tst.pid.d mode=0444
1546 file path=opt/SUNWdtrt/tst/common/safety/tst.ppid.d mode=0444
1547 file path=opt/SUNWdtrt/tst/common/safety/tst.progenyof.d mode=0444
1548 file path=opt/SUNWdtrt/tst/common/safety/tst.random.d mode=0444
1549 file path=opt/SUNWdtrt/tst/common/safety/tst.rw.d mode=0444
1550 file path=opt/SUNWdtrt/tst/common/safety/tst.shortstr.d mode=0444
1551 file path=opt/SUNWdtrt/tst/common/safety/tst.stack.d mode=0444
1552 file path=opt/SUNWdtrt/tst/common/safety/tst.stackdepth.d mode=0444
1553 file path=opt/SUNWdtrt/tst/common/safety/tst.stddev.d mode=0444
1554 file path=opt/SUNWdtrt/tst/common/safety/tst.strchr.d mode=0444
1555 file path=opt/SUNWdtrt/tst/common/safety/tst.strjoin.d mode=0444
1556 file path=opt/SUNWdtrt/tst/common/safety/tst.strstr.d mode=0444
1557 file path=opt/SUNWdtrt/tst/common/safety/tst.strtok.d mode=0444
1558 file path=opt/SUNWdtrt/tst/common/safety/tst.substr.d mode=0444
1559 file path=opt/SUNWdtrt/tst/common/safety/tst.ucaller.d mode=0444
1560 file path=opt/SUNWdtrt/tst/common/safety/tst.uid.d mode=0444
1561 file path=opt/SUNWdtrt/tst/common/safety/tst.unalign.d mode=0444
1562 file path=opt/SUNWdtrt/tst/common/safety/tst.uregs.d mode=0444
1563 file path=opt/SUNWdtrt/tst/common/safety/tst.ustack.d mode=0444
1564 file path=opt/SUNWdtrt/tst/common/safety/tst.ustackdepth.d mode=0444
1565 file path=opt/SUNWdtrt/tst/common/safety/tst.vahole.d mode=0444
1566 file path=opt/SUNWdtrt/tst/common/safety/tst.violentdeath.ksh mode=0444
1567 file path=opt/SUNWdtrt/tst/common/safety/tst.zonename.d mode=0444
1568 file path=opt/SUNWdtrt/tst/common/scalars/err.D_ARR_LOCAL.thisarray.d \
1569 mode=0444
1570 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.selfthis.d \

```



```

1571 mode=0444
1572 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.thisself.d \
1573 mode=0444
1574 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_IDRED.errval.d mode=0444
1575 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dec.err.d \
1576 mode=0444
1577 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupgtype.d \
1578 mode=0444
1579 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupltype.d \
1580 mode=0444
1581 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupptype.d \
1582 mode=0444
1583 file path=opt/SUNWdtrt/tst/common/scalars/err.D_SYNTAX.declare.d mode=0444
1584 file path=opt/SUNWdtrt/tst/common/scalars/err.bigglobal.d mode=0444
1585 file path=opt/SUNWdtrt/tst/common/scalars/err.biglocal.d mode=0444
1586 file path=opt/SUNWdtrt/tst/common/scalars/tst.16kglobal.d mode=0444
1587 file path=opt/SUNWdtrt/tst/common/scalars/tst.16klocal.d mode=0444
1588 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d mode=0444
1589 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d.out mode=0444
1590 file path=opt/SUNWdtrt/tst/common/scalars/tst.localvar.d mode=0444
1591 file path=opt/SUNWdtrt/tst/common/scalars/tst.misc.d mode=0444
1592 file path=opt/SUNWdtrt/tst/common/scalars/tst.self.d mode=0444
1593 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray.d mode=0444
1594 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray2.d mode=0444
1595 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfthis.d mode=0444
1596 file path=opt/SUNWdtrt/tst/common/scalars/tst.this.d mode=0444
1597 file path=opt/SUNWdtrt/tst/common/scalars/tst.thisself.d mode=0444
1598 file path=opt/SUNWdtrt/tst/common/sched/tst.enqueue.d mode=0444
1599 file path=opt/SUNWdtrt/tst/common/sched/tst.oncpu.d mode=0444
1600 file path=opt/SUNWdtrt/tst/common/sched/tst.stackdepth.d mode=0444
1601 file path=opt/SUNWdtrt/tst/common/scripting/err.D_MACRO_UNDEF.invalidargs.d \
1602 mode=0444
1603 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_LVAL.rdonly.d mode=0444
1604 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_WRITE.usepidmacro.d \
1605 mode=0444
1606 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.concat.d mode=0444
1607 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.desc.d mode=0444
1608 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.inval.d mode=0444
1609 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.pid.d mode=0444
1610 file path=opt/SUNWdtrt/tst/common/scripting/tst.D_MACRO_UNUSED.overflow.ksh \
1611 mode=0444
1612 file path=opt/SUNWdtrt/tst/common/scripting/tst.arg0.d mode=0444
1613 file path=opt/SUNWdtrt/tst/common/scripting/tst.arguments.ksh mode=0444
1614 file path=opt/SUNWdtrt/tst/common/scripting/tst.assign.d mode=0444
1615 file path=opt/SUNWdtrt/tst/common/scripting/tst.basic.d mode=0444
1616 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.d mode=0444
1617 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.ksh mode=0444
1618 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.d mode=0444
1619 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.ksh mode=0444
1620 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.d mode=0444
1621 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.ksh mode=0444
1622 file path=opt/SUNWdtrt/tst/common/scripting/tst.pgid.d mode=0444
1623 file path=opt/SUNWdtrt/tst/common/scripting/tst.pid.d mode=0444
1624 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.d mode=0444
1625 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.ksh mode=0444
1626 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.d mode=0444
1627 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.ksh mode=0444
1628 file path=opt/SUNWdtrt/tst/common/scripting/tst.quite.d mode=0444
1629 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.d mode=0444
1630 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.ksh mode=0444
1631 file path=opt/SUNWdtrt/tst/common/scripting/tst.stringmacro.ksh mode=0444
1632 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.d mode=0444
1633 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.ksh mode=0444
1634 file path=opt/SUNWdtrt/tst/common/scripting/tst.trace.d mode=0444
1635 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.d mode=0444
1636 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.ksh mode=0444

```

```

1637 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargets.d mode=0444
1638 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargets.exe mode=0555
1639 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_BADREF.SizeofAssoc.d \
1640 mode=0444
1641 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_UNDEF.UnknownSymbol.d \
1642 mode=0444
1643 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.badstruct.d \
1644 mode=0444
1645 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.d mode=0444
1646 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SYNTAX.SizeofBadType.d \
1647 mode=0444
1648 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofArray.d mode=0444
1649 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofDataTypes.d mode=0444
1650 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofExpression.d mode=0444
1651 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofNULL.d mode=0444
1652 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d mode=0444
1653 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d.out mode=0444
1654 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d mode=0444
1655 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d.out mode=0444
1656 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d mode=0444
1657 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d.out mode=0444
1658 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations1.d \
1659 mode=0444
1660 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations2.d \
1661 mode=0444
1662 file \
1663 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithBreakPo
1664 mode=0444
1665 file \
1666 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithChill.d
1667 mode=0444
1668 file \
1669 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1670 mode=0444
1671 file \
1672 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1673 mode=0444
1674 file \
1675 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithPanic.d
1676 mode=0444
1677 file \
1678 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithRaise.d
1679 mode=0444
1680 file \
1681 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithStop.d
1682 mode=0444
1683 file path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_COMM.AggAftCommit.d \
1684 mode=0444
1685 file \
1686 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithAvg.d \
1687 mode=0444
1688 file \
1689 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithCount.d
1690 mode=0444
1691 file \
1692 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithLquant.
1693 mode=0444
1694 file \
1695 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMax.d \
1696 mode=0444
1697 file \
1698 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMin.d \
1699 mode=0444
1700 file \
1701 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithQuant.d
1702 mode=0444

```

```

1703 file \
1704   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithStddev.
1705   mode=0444
1706 file \
1707   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithSum.d \
1708   mode=0444
1709 file \
1710   path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.CommitAftCommit.d \
1711   mode=0444
1712 file path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.DisjointCommit.d \
1713   mode=0444
1714 file \
1715   path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_DREC.CommitAftDataRec.d
1716   mode=0444
1717 file \
1718   path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.DataRecAftCommit.d
1719   mode=0444
1720 file \
1721   path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.ExitAfterCommit.d \
1722   mode=0444
1723 file path=opt/SUNWdtrt/tst/common/speculation/err.D_EXIT_SPEC.ExitAftSpec.d \
1724   mode=0444
1725 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_MALFORM.NspecExpr.d \
1726   mode=0444
1727 file \
1728   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.HugeNspecValue.
1729   mode=0444
1730 file \
1731   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.InvalidSpecSize
1732   mode=0444
1733 file \
1734   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.NegSpecSize.d \
1735   mode=0444
1736 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PROTO_LEN.SpecNoId.d \
1737   mode=0444
1738 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_COMM.SpecAftCommit.d \
1739   mode=0444
1740 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_DREC.SpecAftDataRec.d \
1741   mode=0444
1742 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_SPEC.SpecAftSpec.d \
1743   mode=0444
1744 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeBufSize.d mode=0444
1745 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeNspec.d mode=0444
1746 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeSpecSize.d mode=0444
1747 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations1.d \
1748   mode=0444
1749 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations2.d \
1750   mode=0444
1751 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitAfterDiscard.d \
1752   mode=0444
1753 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitWithZero.d mode=0444
1754 file path=opt/SUNWdtrt/tst/common/speculation/tst.DataRecAftDiscard.d \
1755   mode=0444
1756 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftCommit.d mode=0444
1757 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDataRec.d \
1758   mode=0444
1759 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDiscard.d \
1760   mode=0444
1761 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardWithZero.d mode=0444
1762 file path=opt/SUNWdtrt/tst/common/speculation/tst.ExitAftDiscard.d mode=0444
1763 file path=opt/SUNWdtrt/tst/common/speculation/tst.NoSpecBuffer.d mode=0444
1764 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations1.d \
1765   mode=0444
1766 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations2.d \
1767   mode=0444
1768 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations3.d \

```

```

1769   mode=0444
1770 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculateWithRandom.d \
1771   mode=0444
1772 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationCommit.d \
1773   mode=0444
1774 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationDiscard.d \
1775   mode=0444
1776 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationID.d mode=0444
1777 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationWithZero.d \
1778   mode=0444
1779 file path=opt/SUNWdtrt/tst/common/speculation/tst.TwoSpecBuffers.d mode=0444
1780 file path=opt/SUNWdtrt/tst/common/speculation/tst.negcommit.d mode=0444
1781 file path=opt/SUNWdtrt/tst/common/speculation/tst.negspec.d mode=0444
1782 file path=opt/SUNWdtrt/tst/common/speculation/tst.zeroseize.d mode=0444
1783 file path=opt/SUNWdtrt/tst/common/stability/err.D_ATTR_MIN.MinAttributes.d \
1784   mode=0444
1785 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_PROTO.bad.d mode=0444
1786 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_SIZE.d mode=0444
1787 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_FRAMES.bad.d mode=0444
1788 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_PROTO.bad.d mode=0444
1789 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_STRSIZE.bad.d mode=0444
1790 file path=opt/SUNWdtrt/tst/common/stack/tst.default.d mode=0444
1791 file path=opt/SUNWdtrt/tst/common/stackdepth/tst.default.d mode=0444
1792 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.d mode=0444
1793 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.exe mode=0555
1794 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.d mode=0444
1795 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.exe mode=0555
1796 file path=opt/SUNWdtrt/tst/common/strlen/tst.strlen1.d mode=0444
1797 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooLarge.d mode=0444
1798 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooSmall.d mode=0444
1799 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d mode=0444
1800 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d.out mode=0444
1801 file path=opt/SUNWdtrt/tst/common/struct/err.D_ADDR_OF_VAR.StructPointer.d \
1802   mode=0444
1803 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon.d \
1804   mode=0444
1805 file \
1806   path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon1.d \
1807   mode=0444
1808 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.circular.d \
1809   mode=0444
1810 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order.d \
1811   mode=0444
1812 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order2.d \
1813   mode=0444
1814 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.recursive.d \
1815   mode=0444
1816 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.simple.d \
1817   mode=0444
1818 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_VOIDOBJ.baddec.d mode=0444
1819 file path=opt/SUNWdtrt/tst/common/struct/err.D_PROTO_ARG.DupStructAssoc.d \
1820   mode=0444
1821 file path=opt/SUNWdtrt/tst/common/struct/tst.StructAssoc.d mode=0444
1822 file path=opt/SUNWdtrt/tst/common/struct/tst.StructDataTypes.d mode=0444
1823 file path=opt/SUNWdtrt/tst/common/struct/tst.StructInside.d mode=0444
1824 file path=opt/SUNWdtrt/tst/common/struct/tst.clauselocal.d mode=0444
1825 file path=opt/SUNWdtrt/tst/common/struct/tst.clauselocal.d.out mode=0444
1826 file path=opt/SUNWdtrt/tst/common/sugar/tst.else.d mode=0444
1827 file path=opt/SUNWdtrt/tst/common/sugar/tst.if.d mode=0444
1828 file path=opt/SUNWdtrt/tst/common/sugar/tst.if2.d mode=0444
1829 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_before_after.d mode=0444
1830 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_nested.d mode=0444
1831 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon.d mode=0444
1832 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon2.d mode=0444
1833 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.d mode=0444
1834 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.exe mode=0555

```

```

1835 file path=opt/SUNWdtrt/tst/common/syscall/tst.openret.ksh mode=0444
1836 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.d mode=0444
1837 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.exe mode=0555
1838 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.d mode=0444
1839 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.exe mode=0555
1840 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZERO.tick.d mode=0444
1841 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonens.d mode=0444
1842 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonensec.d mode=0444
1843 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneus.d mode=0444
1844 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneusec.d mode=0444
1845 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickarg0.d mode=0444
1846 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d mode=0444
1847 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d.out mode=0444
1848 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d mode=0444
1849 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d.out mode=0444
1850 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d mode=0444
1851 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d.out mode=0444
1852 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d mode=0444
1853 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d.out mode=0444
1854 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d mode=0444
1855 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d.out mode=0444
1856 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d mode=0444
1857 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d.out mode=0444
1858 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d mode=0444
1859 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d.out mode=0444
1860 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d mode=0444
1861 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d.out mode=0444
1862 file path=opt/SUNWdtrt/tst/common/trace/err.D_PROTO_LEN.bad.d mode=0444
1863 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_AGG.bad.d mode=0444
1864 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_VOID.bad.d mode=0444
1865 file path=opt/SUNWdtrt/tst/common/trace/tst.dyn.d mode=0444
1866 file path=opt/SUNWdtrt/tst/common/trace/tst.misc.d mode=0444
1867 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d mode=0444
1868 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d.out mode=0444
1869 file path=opt/SUNWdtrt/tst/common/trace/tst.string.d mode=0444
1870 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_ARG.badsize.d mode=0444
1871 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_LEN.toofew.d mode=0444
1872 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ADDR.badaddr.d \
1873 mode=0444
1874 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ARGS.d mode=0444
1875 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_DYN_SIZE.d mode=0444
1876 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.negsize.d \
1877 mode=0444
1878 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.zerosize.d \
1879 mode=0444
1880 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d mode=0444
1881 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d.out mode=0444
1882 file path=opt/SUNWdtrt/tst/common/tracemem/tst.rootvtp.d mode=0444
1883 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d mode=0444
1884 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d.out mode=0444
1885 file \
1886 path=opt/SUNWdtrt/tst/common/translators/err.D_DECL_TYPERED.BadTransDecl.d \
1887 mode=0444
1888 file \
1889 path=opt/SUNWdtrt/tst/common/translators/err.D_OP_INCOMPLETE.NonExistentInpu
1890 mode=0444
1891 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl1.d \
1892 mode=0444
1893 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl3.d \
1894 mode=0444
1895 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl4.d \
1896 mode=0444
1897 file \
1898 path=opt/SUNWdtrt/tst/common/translators/err.D_TYPE_MEMBER.NonExistentInput2
1899 mode=0444
1900 file \

```

```

1901 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_INCOMPAT.BadInputType1.
1902 mode=0444
1903 file \
1904 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_MEMB.NonExistentOutput2
1905 mode=0444
1906 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_NONE.BadTransDecl6.d \
1907 mode=0444
1908 file \
1909 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_REDECL.RepeatTransDecl.
1910 mode=0444
1911 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransDecl8.d \
1912 mode=0444
1913 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransInt.d \
1914 mode=0444
1915 file \
1916 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.NonExistentOutput1.
1917 mode=0444
1918 file path=opt/SUNWdtrt/tst/common/translators/tst.CircularTransDecl.d \
1919 mode=0444
1920 file path=opt/SUNWdtrt/tst/common/translators/tst.EmptyTransDecl.d mode=0444
1921 file path=opt/SUNWdtrt/tst/common/translators/tst.ForwardTag.d mode=0444
1922 file path=opt/SUNWdtrt/tst/common/translators/tst.InputAliasTrans.d mode=0444
1923 file path=opt/SUNWdtrt/tst/common/translators/tst.InputIntTrans.d mode=0444
1924 file path=opt/SUNWdtrt/tst/common/translators/tst.OutputAliasTrans.d mode=0444
1925 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialDereferencing.d \
1926 mode=0444
1927 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialOutputTransDefn.d \
1928 mode=0444
1929 file path=opt/SUNWdtrt/tst/common/translators/tst.ProcModelTrans.d mode=0444
1930 file path=opt/SUNWdtrt/tst/common/translators/tst.RepeatDeclaration.d \
1931 mode=0444
1932 file path=opt/SUNWdtrt/tst/common/translators/tst.SimultaneousTranslators.d \
1933 mode=0444
1934 file path=opt/SUNWdtrt/tst/common/translators/tst.StructureAssignment.d \
1935 mode=0444
1936 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh \
1937 mode=0444
1938 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh.out \
1939 mode=0444
1940 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh \
1941 mode=0444
1942 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh.out \
1943 mode=0444
1944 file path=opt/SUNWdtrt/tst/common/translators/tst.TransNonPointer.d mode=0444
1945 file path=opt/SUNWdtrt/tst/common/translators/tst.TransOutputPointer.d \
1946 mode=0444
1947 file path=opt/SUNWdtrt/tst/common/translators/tst.TransPointer.d mode=0444
1948 file path=opt/SUNWdtrt/tst/common/translators/tst.TranslateSelf.d mode=0444
1949 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionInputTrans.d mode=0444
1950 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionOutputTrans.d mode=0444
1951 file path=opt/SUNWdtrt/tst/common/typedef/err.D_DECL_IDRED.DupTypeDef.d \
1952 mode=0444
1953 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.BadExistingTypeDef.d \
1954 mode=0444
1955 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.TypeDefInClause.d \
1956 mode=0444
1957 file path=opt/SUNWdtrt/tst/common/typedef/tst.ChainTypeDef.d mode=0444
1958 file path=opt/SUNWdtrt/tst/common/typedef/tst.TypeDefDataAssign.d mode=0444
1959 file path=opt/SUNWdtrt/tst/common/types/err.D_CAST_INVAL.badcast.d mode=0444
1960 file path=opt/SUNWdtrt/tst/common/types/err.D_CG_DYN.ResultDynType.d mode=0444
1961 file path=opt/SUNWdtrt/tst/common/types/err.D_CHR_OFLOW.charconst.d mode=0444
1962 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_BADCLASS.bad.d mode=0444
1963 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_CHARATTR.badtype3.d \
1964 mode=0444
1965 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype4.d mode=0444
1966 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype5.d mode=0444

```

```

1967 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdatrr1.d mode=0444
1968 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdatrr2.d mode=0444
1969 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdatrr3.d mode=0444
1970 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdatrr4.d mode=0444
1971 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdatrr5.d mode=0444
1972 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENCONST.badeval.d mode=0444
1973 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enoflow.d mode=0444
1974 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enuflow.d mode=0444
1975 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_SCOPE.scopeop.d mode=0444
1976 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_USELESS.baddec.d mode=0444
1977 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ACT.badcond.d mode=0444
1978 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ARITH.badoperand.d mode=0444
1979 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INCOMPAT.badassign.d \
1980 mode=0444
1981 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badbitop.d mode=0444
1982 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badshift.d mode=0444
1983 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badcond.d mode=0444
1984 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badincop.d mode=0444
1985 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badlogop.d mode=0444
1986 file path=opt/SUNWdtrt/tst/common/types/err.D_PROTO_LEN.badcond1.d mode=0444
1987 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badenum.d mode=0444
1988 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badid.d mode=0444
1989 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badstruct.d mode=0444
1990 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype1.d mode=0444
1991 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype2.d mode=0444
1992 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupenum.d mode=0444
1993 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupstruct.d mode=0444
1994 file path=opt/SUNWdtrt/tst/common/types/err.D_XLATE_REDECL.ResultDynType.d \
1995 mode=0444
1996 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt1.d mode=0444
1997 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt2.d mode=0444
1998 file path=opt/SUNWdtrt/tst/common/types/tst.assignops.d mode=0444
1999 file path=opt/SUNWdtrt/tst/common/types/tst.badshiftops.d mode=0444
2000 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d mode=0444
2001 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d.out mode=0444
2002 file path=opt/SUNWdtrt/tst/common/types/tst.bitops.d mode=0444
2003 file path=opt/SUNWdtrt/tst/common/types/tst.charconstants.d mode=0444
2004 file path=opt/SUNWdtrt/tst/common/types/tst.complex.d mode=0444
2005 file path=opt/SUNWdtrt/tst/common/types/tst.condexpr.d mode=0444
2006 file path=opt/SUNWdtrt/tst/common/types/tst.const.d mode=0444
2007 file path=opt/SUNWdtrt/tst/common/types/tst.constants.d mode=0444
2008 file path=opt/SUNWdtrt/tst/common/types/tst.conv.d mode=0444
2009 file path=opt/SUNWdtrt/tst/common/types/tst.enum.d mode=0444
2010 file path=opt/SUNWdtrt/tst/common/types/tst.intincop.d mode=0444
2011 file path=opt/SUNWdtrt/tst/common/types/tst.intops.d mode=0444
2012 file path=opt/SUNWdtrt/tst/common/types/tst.inttypes.d mode=0444
2013 file path=opt/SUNWdtrt/tst/common/types/tst.ptrincop.d mode=0444
2014 file path=opt/SUNWdtrt/tst/common/types/tst.ptrops.d mode=0444
2015 file path=opt/SUNWdtrt/tst/common/types/tst.relenum.d mode=0444
2016 file path=opt/SUNWdtrt/tst/common/types/tst.relistring.d mode=0444
2017 file path=opt/SUNWdtrt/tst/common/types/tst.shiftops.d mode=0444
2018 file path=opt/SUNWdtrt/tst/common/types/tst.stringconstants.d mode=0444
2019 file path=opt/SUNWdtrt/tst/common/types/tst.struct.d mode=0444
2020 file path=opt/SUNWdtrt/tst/common/types/tst.typedef.d mode=0444
2021 file path=opt/SUNWdtrt/tst/common/types/tst.unaryop.d mode=0444
2022 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid.d mode=0444
2023 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid2.d mode=0444
2024 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid3.d mode=0444
2025 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype.ksh mode=0444
2026 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype2.ksh mode=0444
2027 file path=opt/SUNWdtrt/tst/common/uctf/err.user64mode.ksh mode=0444
2028 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.exe mode=0555
2029 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.ksh mode=0444
2030 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.exe mode=0555
2031 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh mode=0444
2032 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh.out mode=0444

```

```

2033 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.exe mode=0555
2034 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.ksh mode=0444
2035 file path=opt/SUNWdtrt/tst/common/uctf/tst.linkmap.ksh mode=0444
2036 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprint.ksh mode=0444
2037 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprintarg.ksh mode=0444
2038 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.exe mode=0555
2039 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh mode=0444
2040 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh.out mode=0444
2041 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtypetarg.ksh mode=0444
2042 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh mode=0444
2043 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh.out mode=0444
2044 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh mode=0444
2045 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh.out mode=0444
2046 file path=opt/SUNWdtrt/tst/common/union/err.D_ADDROF_VAR.UnionPointer.d \
2047 mode=0444
2048 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon.d \
2049 mode=0444
2050 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon1.d \
2051 mode=0444
2052 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.circular.d \
2053 mode=0444
2054 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.order.d \
2055 mode=0444
2056 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.recursive.d \
2057 mode=0444
2058 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.simple.d \
2059 mode=0444
2060 file path=opt/SUNWdtrt/tst/common/union/err.D_PROTO_ARG.DupUnionAssoc.d \
2061 mode=0444
2062 file path=opt/SUNWdtrt/tst/common/union/tst.UnionAssoc.d mode=0444
2063 file path=opt/SUNWdtrt/tst/common/union/tst.UnionDataTypes.d mode=0444
2064 file path=opt/SUNWdtrt/tst/common/union/tst.UnionInside.d mode=0444
2065 file path=opt/SUNWdtrt/tst/common/usdt/tst.andpid.ksh mode=0444
2066 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.d mode=0444
2067 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.exe mode=0555
2068 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.d mode=0444
2069 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.exe mode=0555
2070 file path=opt/SUNWdtrt/tst/common/usdt/tst.badguess.ksh mode=0444
2071 file path=opt/SUNWdtrt/tst/common/usdt/tst.corruptenv.ksh mode=0444
2072 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh mode=0444
2073 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh.out mode=0444
2074 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh mode=0444
2075 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh.out mode=0444
2076 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose3.ksh mode=0444
2077 file path=opt/SUNWdtrt/tst/common/usdt/tst.eliminate.ksh mode=0444
2078 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh mode=0444
2079 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh.out mode=0444
2080 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh mode=0444
2081 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh.out mode=0444
2082 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh mode=0444
2083 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh.out mode=0444
2084 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh mode=0444
2085 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh.out mode=0444
2086 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.exe mode=0555
2087 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.ksh mode=0444
2088 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess32.ksh mode=0444
2089 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess64.ksh mode=0444
2090 file path=opt/SUNWdtrt/tst/common/usdt/tst.header.ksh mode=0444
2091 file path=opt/SUNWdtrt/tst/common/usdt/tst.include.ksh mode=0444
2092 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe.exe mode=0555
2093 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe1.ksh mode=0444
2094 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe2.ksh mode=0444
2095 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkpriv.ksh mode=0444
2096 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkpriv.ksh mode=0444
2097 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh mode=0444
2098 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh.out mode=0444

```

```
2099 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh mode=0444
2100 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh.out mode=0444
2101 file path=opt/SUNWdtrt/tst/common/usdt/tst.nodtrace.ksh mode=0444
2102 file path=opt/SUNWdtrt/tst/common/usdt/tst.noprobes.ksh mode=0444
2103 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreap.ksh mode=0444
2104 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreapring.ksh mode=0444
2105 file path=opt/SUNWdtrt/tst/common/usdt/tst.onlyenabled.ksh mode=0444
2106 file path=opt/SUNWdtrt/tst/common/usdt/tst.reap.ksh mode=0444
2107 file path=opt/SUNWdtrt/tst/common/usdt/tst.reeval.ksh mode=0444
2108 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh mode=0444
2109 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh.out mode=0444
2110 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh mode=0444
2111 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh.out mode=0444
2112 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh mode=0444
2113 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh.out mode=0444
2114 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh mode=0444
2115 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh.out mode=0444
2116 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.d mode=0444
2117 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.exe mode=0555
2118 file path=opt/SUNWdtrt/tst/common/ustack/tst.depth.ksh mode=0444
2119 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.exe mode=0555
2120 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.ksh mode=0444
2121 file path=opt/SUNWdtrt/tst/common/vars/tst.gid.d mode=0444
2122 file path=opt/SUNWdtrt/tst/common/vars/tst.nullassign.d mode=0444
2123 file path=opt/SUNWdtrt/tst/common/vars/tst.ppid.d mode=0444
2124 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh mode=0444
2125 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh.out mode=0444
2126 file path=opt/SUNWdtrt/tst/common/vars/tst.uid.d mode=0444
2127 file path=opt/SUNWdtrt/tst/common/vars/tst.walltimestamp.d mode=0444
2128 file path=opt/SUNWdtrt/tst/common/version/tst.1.0.d mode=0444
2129 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.basic.ksh mode=0444
2130 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.hvmenable.ksh mode=0444
2131 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.memenable.ksh mode=0444
2132 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedargs.ksh mode=0444
2133 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedenable.ksh \
2134     mode=0444
2135 legacy pkg=SUNWdtrt category=internal \
2136     desc="DTrace Test Suite Internal Distribution" \
2137     hotline="Contact the DTrace discussion forum" name="DTrace Test Suite"
2138 license cr_Sun license=cr_Sun
2139 license lic_CDDL license=lic_CDDL
2140 depend fmri=runtime/java type=require
2141 depend fmri=runtime/java/runtime64 type=require
```

new/usr/src/test/os-tests/tests/file-locking/Makefile

1

```
*****
1575 Thu Feb 22 01:20:16 2018
new/usr/src/test/os-tests/tests/file-locking/Makefile
9128 cw(lonbld) shouldn't be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2016 Joyent, Inc.
14 #
15 #
16 include $(SRC)/cmd/Makefile.cmd
17 include $(SRC)/test/Makefile.com
18 #
19 OBJECTS=util.o runtests.32.o runtests.64.o acquire-lock.32.o acquire-lock.64.o
19 UTILS = util.c
20 #
21 PROGS = \
22     runtests \
23     acquire-lock
24 #
25 CSTD = $(CSTD_GNU99)
26 #
27 SRCS = $(PROGS:%=%.c) $(UTILS:%.o=%.c)
27 SRCS = $(PROGS:%=%.c) $(UTILS)
28 PROGS32 = $(PROGS:%=%.32)
29 PROGS64 = $(PROGS:%=%.64)
30 #
31 LINTS = $(PROGS:%=%.ln)
32 LINTFLAGS += -erroff=E_NAME_DEF_NOT_USED2
33 LINTFLAGS += -erroff=E_NAME_USED_NOT_DEF2
34 #
35 ROOTOPTDIR = $(ROOT)/opt/os-tests/tests/file-locking
36 ROOTOPTPROGS = $(PROGS32:%=$(ROOTOPTDIR)/%) \
37     $(PROGS64:%=$(ROOTOPTDIR)/%)
38 #
39 .KEEP_STATE:
40 #
41 install: $(ROOTOPTPROGS)
42 #
43 all: $(PROGS32) $(PROGS64)
44 #
45 lint: $(LINTS)
46 #
47 clean:
48     -rm $(PROGS32) $(PROGS64)
49 #
50 $(ROOTOPTPROGS): $(PROGS32) $(PROGS64) $(ROOTOPTDIR)
51 #
52 $(ROOTOPTDIR):
```

new/usr/src/test/os-tests/tests/file-locking/Makefile

2

```
53     $(INS.dir)
54 #
55 $(ROOTOPTDIR)/%: %
56     $(INS.file)
57 #
58 $(ROOTOPTDIR)/%: %.ksh
59     $(INS.rename)
60 #
61 %.ln: %.c
62     $(LINT.c) $< $(UTILS) $(LDLIBS)
63 #
64 %.64.o: %.c
65     $(COMPILE64.c) $< -o $@
66 #
67 %.32.o: %.c
68     $(COMPILE.c) $< -o $@
69 #
70 %.64: %.64.o util.64.o
71     $(LINK64.c) -o $@ $< util.64.o $(LDLIBS64)
72     $(LINK.c) $< $(UTILS) $(LDLIBS)
73     $(POST_PROCESS)
74 #
74 %.32: %.32.o util.32.o
75     $(LINK.c) -o $@ $< util.32.o $(LDLIBS)
76     $(LINK.c) $< $(UTILS) $(LDLIBS)
77     $(POST_PROCESS)
78 #
79 clobber:
80     $(RM) $(PROGS32) $(PROGS64)
81 #
82 FRC:
```

new/usr/src/tools/cw/Makefile

1

```
*****
1856 Thu Feb 22 01:20:17 2018
new/usr/src/tools/cw/Makefile
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 PROG      = cw

28 MAN1ONBLDFILES= cw.lonbld

30 include ../Makefile.tools

32 # Bootstrap problem -- we have to build cw before we can use it
33 i386_CC=    $(SPRO_VROOT)/bin/cc
34 sparc_CC=   $(SPRO_VROOT)/bin/cc
35 $(__GNUC)i386_CC= $(GCC_ROOT)/bin/gcc
36 $(__GNUC)sparc_CC= $(GCC_ROOT)/bin/gcc

38 CFLAGS += $(CCVERBOSE)

40 # Override CFLAGS. This is needed only for bootstrap of cw.
41 $(__GNUC)CFLAGS= -O -D_sun -Wall -Wno-unknown-pragmas -Werror \
42                -std=gnu99 -nodefaultlibs
43                -std=gnu89 -nodefaultlibs
44 $(__SUNC)CFLAGS= -xspace -Xa -xildoff -errtags=yes -errwarn=%all \
45                -xc99=%all -W0,-xglobalstatic -v
46                -xc99=%none -W0,-xglobalstatic -v

47 $(__GNUC)LDLIBS += -lc
48 $(__GNUC)LDFLAGS= $(MAPFILE.NES:%=-Wl,-M%)

50 CPPFLAGS += -DDEFAULT_CC_DIR="$(SPRO_VROOT)/bin"
51 CPPFLAGS += -DDEFAULT_GCC_DIR="$(GCC_ROOT)/bin"
52 CPPFLAGS += -DDEFAULT_CPLUSPLUS_DIR="$(SPRO_VROOT)/bin"
```

new/usr/src/tools/cw/Makefile

2

```
53 CPPFLAGS += -DDEFAULT_GPLUSPLUS_DIR="$(GCC_ROOT)/bin"

51 $(ROOTONBLDMAN1ONBLDFILES) := FILEMODE= 644

53 # Assume we don't have the install.bin available yet
54 INS.file= $(RM) $@; $(CP) $< $(@); $(CHMOD) $(FILEMODE) $@

56 .KEEP_STATE:

58 all: $(PROG) $(MAN1ONBLDFILES)

60 install: all .WAIT $(ROOTONBLDMACHPROG) $(ROOTONBLDMAN1ONBLDFILES)

62 lint: lint_PROG

64 clean:

66 include ../Makefile.targ
```

```

*****
6240 Thu Feb 22 01:20:18 2018
new/usr/src/tools/cw/cw.lonbld
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
1  .\"
2  .\" CDDL HEADER START
3  .\"
4  .\" The contents of this file are subject to the terms of the
5  .\" Common Development and Distribution License (the "License").
6  .\" You may not use this file except in compliance with the License.
7  .\"
8  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  .\" or http://www.opensolaris.org/os/licensing.
10 .\" See the License for the specific language governing permissions
11 .\" and limitations under the License.
12 .\"
13 .\" When distributing Covered Code, include this CDDL HEADER in each
14 .\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 .\" If applicable, add the following below this CDDL HEADER, with the
16 .\" fields enclosed by brackets "[" replaced with your own identifying
17 .\" information: Portions Copyright [yyyy] [name of copyright owner]
18 .\"
19 .\" CDDL HEADER END
20 .\"
21 .\" Copyright 2010 Sun Microsystems, Inc. All rights reserved.
22 .\" Use is subject to license terms.
23 .\"
24 .Dd February 10, 2018
25 .Dt CW LONBLD
26 .Os
27 .Sh NAME
28 .Nm cw
29 .Nd invoke one or more compilers with argument translation
30 .Sh SYNOPSIS
31 .Nm cw
32 .Op Fl C
33 .Op Fl -versions
34 .Op Fl -noecho
35 .Fl -primary Ar compiler
36 .Op Fl -shadow Ar compiler ...
37 .Fl -
38 .Ar compiler args ...
39 .Sh DESCRIPTION
40 .Nm cw
41 is a facility for invoking one or more compilers, providing translation from
42 Sun style arguments as appropriate.
43 This allows the use of arbitrary compilers without the need to alter large
44 numbers of makefiles.
45 A mode called shadow compilation invokes multiple compilers so that warnings
46 and errors may be obtained from both.
47 See
48 .Sx SHADOW COMPILATION
49 for details.
50 This version of cw supports compilers with both Sun Studio 12 and GCC-style
51 command lines.
52 .Sh ARGUMENTS
53 Both the
54 .Fl -primary

```

```

55 and
56 .Fl -shadow
57 parameters take a
58 .Em compiler specification .
59 This is a comma-separated list of the form
60 .Ar name,executable,style
61 Where
62 .Ar name
63 is a name for the compiler,
64 .Ar executable
65 is the full path to the compiler executable, and
66 .Ar style
67 is the style of command-line options the compiler expects, either
68 .Em sun
69 or
70 .Em gnu .
71 .Bl -tag -width indent
72 .It Fl -primary Ar compiler
73 Specify the compiler to be used primarily (that which is used for link-editing
74 and pre-processing, and whos objects we deliver).
75 .It Fl -shadow Ar compiler
76 Specify a shadow compiler, which builds sources for the sake of checking code
77 quality and compatibility, but has its output discarded.
78 .It Fl -noecho
79 Do not echo the actual command line of any compilers invoked.
80 .It Fl -versions
81 Request from each configured primary and shadow compiler its version
82 information.
83 .It Fl C
84 The sources being compiled are C++. This is necessary as it affects the
85 translation of compiler arguments.
86 .It Fl -
87 Arguments intended for the compilers themselves must be separated from those
88 of
89 .Nm cw
90 by a
91 .Fl - .
92 .It Fl _name=
93 .It Fl _style=
94 Parameters intended for the compiler be guarded with options of the form
95 .Fl _name=
96 and
97 .Fl _style=
98 Where
99 .Em name
100 and
101 .Em style
102 are those passed to
103 .Fl -primary
104 and
105 .Fl -shadow
106 this allows certain flags to be passed only to certain classes of compiler.
107 .Pp
108 For historical reasons, the
109 .Fl _style=
110 option is also translated such that a style of
111 .Em sun
112 may use the flag
113 .Fl _cc=
114 and a style of
115 .Em gnu
116 may use the flag
117 .Fl _gcc= ,
118 and when the
119 .Fl C
120 option is given and C++ is in use the style of

```



```

121 .Em sun
122 may use the flag
123 .Fl _CC=
124 and the style of
125 .Em gnu
126 may use the flag
127 .Fl _g++= .
128 .El
129 .Sh SHADOW COMPILATION
130 If
131 .Fl -shadow
132 compilers are specified
133 .Nm cw
134 will invoke shadow compiler, with the outputs modified (as well as any
135 translation for compiler style) as follows:
136 .Bl -enum
137 .It
138 If neither of
139 .Fl c ,
140 .Fl s
141 appears in the argument list (that is, linking is attempted or only the
142 pre-processor is invoked), the shadow compilers will not be invoked.
143 .It
144 If the
145 .Fl o Ar filename
146 option was provided, with or without a separating space, it will be replaced wit
147 .Fl o Ar tempfile
148 .It
149 If the option
150 .Fl o
151 was not provided,
152 .Fl o Ar tempfile
153 will be added to the end of the argument list used to invoke
154 the shadow compilers.
155 .El
156 When shadow compilation is in effect,
157 .Nm cw
158 writes to standard error each compiler's standard error output following its
159 argument list.
160 Messages from the compilers will not be interleaved.
161 If
162 .Nm cw
24 .TH CW 1ONBLD "Mar 22, 2010"
25 .SH NAME
26 .I cw
27 \- invoke one or more compilers with argument translation
28 .SH SYNOPSIS
29 \fBcw {-_cc/_gcc/_CC/_g++} [options] [compiler args...]\fP
30 .SH DESCRIPTION
31 .LP
32 .I cw
33 is a facility for invoking one or more compilers, providing
34 translation from Sun Studio arguments as appropriate. This allows
35 the use of arbitrary compilers without the need to alter large
36 numbers of makefiles. A mode called shadow compilation, the default,
37 invokes two different compilers so that warnings and errors may be
38 obtained from both. See SHADOW COMPILATION for details. This version
39 of cw supports Sun Studio 10 and 11 and gcc 3.4.3 as shipped in
40 Solaris.
41 .SH ARGUMENTS
42 .LP
43 Exactly one of the following arguments is required, and must appear
44 immediately following \fBcw\fP in the argument list:
45 .TP 4
46 .B \-_cc
47 Select Sun Studio C (cc) as the primary compiler.

```

```

48 .TP 4
49 .B \-_gcc
50 Select GNU C (gcc) as the primary compiler.
51 .TP 4
52 .B \-_CC
53 Select Sun Studio C++ (CC) as the primary compiler.
54 .TP 4
55 .B \-_g++
56 Select GNU C++ (g++) as the primary compiler.
57 .SH OPTIONS
58 .LP
59 The following options are supported:
60 .TP 4
61 .B \-_compiler
62 Do not invoke any compiler; instead, write to standard output the
63 full path to the primary compiler that would be invoked. If this
64 option is given, it must appear immediately following the compiler
65 selection argument, and all other options will be disregarded.
66 .TP 4
67 .B \-_versions
68 Print the cw version, followed in turn by the path to and versions
69 of each compiler which would be invoked. If this option is given,
70 it must appear immediately following the compiler selection argument,
71 and all other options will be disregarded.
72 .TP 4
73 .B \-_noecho
74 Normally, cw writes to standard error the actual compiler commands
75 invoked, each prefixed with '+'. This option suppresses these
76 messages.
77 .TP 4
78 .B \-_cc=,\-_gcc=,\-_CC=,\-_g++=
79 Pass compiler-dependent options. See ARGUMENT TRANSLATION.
80 .SH SHADOW COMPILATION
81 .LP
82 Unless otherwise directed by environment variables (see ENVIRONMENT),
83 \fBcw\fP will invoke both the compiler specified by its first argument
84 (primary) and a designated alternate compiler (shadow). The primary
85 compiler will be invoked with the arguments given to cw, translated
86 as described in ARGUMENT TRANSLATION. The secondary compiler will be
87 invoked in parallel with the primary (but see ENVIRONMENT); its
88 arguments will likewise be translated. However, the shadow compiler's
89 arguments will also be modified as follows:
90 .TP 4
91 .B 1.
92 If none of \fI-c\fP, \fI-E\fP, \fI-P\fP, or \fI-S\fP appears in the
93 argument list (that is, linking is attempted), the shadow compiler
94 will not be invoked. This is because the objects built with that
95 compiler which would be linked have been previously discarded.
96 .TP 4
97 .B 2.
98 If an option of the form \fI-ofilename\fP was provided, it will be
99 replaced by two options of the form \fI-o\fP \fItempfile\fP.
100 .TP 4
101 .B 3.
102 If the option \fI-o\fP was provided, its following argument will be
103 replaced by \fItempfile\fP.
104 .TP 4
105 .B 4.
106 If neither of the above options was provided, two options of the
107 form \fI-o\fP \fItempfile\fP will be added to the end of the argument
108 list used to invoke the shadow compiler.
109 .LP
110 In all three cases, \fItempfile\fP is the name of a temporary file
111 which will be removed by \fBcw\fP before it terminates.
112 .LP
113 When shadow compilation is in effect, \fBcw\fP writes to standard error

```

```

114 each compiler's standard error output following its argument list.
115 Messages from the two compilers will not be interleaved. If \fBcw\fp
116 is used to invoke the preprocessor and no output location is specified,
117 .Nm cw
118 will write to standard output the primary compiler's standard output.
119 .Pp
120 Because the Sun compilers write intermediate objects to fixed
121 \fBcw\fp will write to standard output the primary compiler's
122 standard output, and the secondary compiler's standard output will
123 be discarded.
124 .LP
125 Because the Studio compilers write intermediate objects to fixed
126 filenames in the current directory when instructed to compile and
127 link multiple source files via a single command line, it would be
128 unsafe to invoke more than one compiler in this fashion.
129 Therefore
130 .Nm cw
131 does not accept multiple source files unless the preprocessor is to be
132 invoked.
133 An attempt to invoke
134 .Nm cw
135 in this manner will result in an error.
136 .Sh ARGUMENT TRANSLATION
137 If the compiler to be invoked is a GNU-style C or C++ compiler, a set of
138 unsafe to invoke more than one compiler in this fashion. Therefore
139 \fBcw\fp does not accept multiple source files unless the
140 preprocessor is to be invoked. An attempt to invoke \fBcw\fp in
141 this manner will result in an error.
142 .SH ARGUMENT TRANSLATION
143 .LP
144 The arguments provided to \fBcw\fp will be passed through to each
145 compiler invoked, altered as follows:
146 .TP 4
147 .B 1.
148 Options intended for \fBcw\fp itself are removed.
149 .TP 4
150 .B 2.
151 The shadow compiler's arguments are modified as described in
152 SHADOW COMPILATION so that its output is discarded.
153 .TP 4
154 .B 3.
155 If the compiler to be invoked is a GNU C or C++ compiler, a set of
156 default flags is added to the beginning of the argument list, and the
157 remaining arguments are translated to their closest appropriate
158 semantic equivalents and passed in the same order as their
159 counterparts given to
160 .Nm cw .
161 See the comments at the head of
162 .Pa usr/src/tools/cw/cw.c
163 for a detailed list of translations.
164 .Sh ENVIRONMENT
165 .Bl -tag -width indent
166 .It CW_SHADOW_SERIAL
167 If this variable is set in the environment, invoke the primary compiler, wait
168 for it to complete, then invoke the shadow compilers.
169 Normally the primary and shadow compilers are invoked in parallel.
170 .It CW_NO_EXEC
171 counterparts given to \fBcw\fp. Arguments which begin with
172 \fI-gcc=\fp or \fI-g++=\fp will be passed in place with the
173 leading portion removed. Arguments which begin with \fI-cc=\fp or
174 \fI-CC=\fp will be discarded. See the comments at the head of
175 \fIusr/src/tools/cw/cw.c\fp for a detailed list of translations.
176 .TP 4
177 .B 4.
178 If the compiler to be invoked is a Studio C or C++ compiler, the
179 remaining arguments are passed unmodified and in the same order as

```

```

154 given to \fBcw\fp. Arguments which begin with \fI-cc=\fp or
155 \fI-CC=\fp will be passed in place with the leading portion removed.
156 Arguments which begin with \fI-gcc=\fp or \fI-g++=\fp will be
157 discarded.
158 .SH ENVIRONMENT
159 .TP 4
160 .B CW_NO_SHADOW
161 If this variable is set in the environment, invoke only the
162 primary compiler.
163 .TP 4
164 .B CW_SHADOW_SERIAL
165 If this variable is set in the environment, invoke the primary compiler,
166 wait for it to complete, then invoke the shadow compiler. Normally
167 the two compilers are invoked in parallel. If CW_NO_SHADOW is set,
168 this has no effect.
169 .TP 4
170 .B CW_NO_EXEC
171 If this variable is set in the environment, write the usual output to
172 standard error but do not actually invoke any compiler.
173 This is useful for debugging the translation engine.
174 .El
175 .Sh EXIT STATUS
176 standard error but do not actually invoke any compiler. This is
177 useful for debugging the translation engine.
178 .TP 4
179 .B CW_CC, CW_CPLUSPLUS
180 If these variables are set in the environment, they specify the full
181 pathname for the Studio C and C++ compilers, respectively.
182 .TP 4
183 .B CW_CC_DIR, CW_CPLUSPLUS_DIR, SPRO_VROOT, SPRO_ROOT, BUILD_TOOLS
184 If CW_CC or CW_CPLUSPLUS are not set,
185 these variables define the search path for Studio compilers as
186 follows:
187 .B
188 If CW_CC_DIR is set, the Studio C compiler in \fI$CW_CC_DIR\fp
189 will be used. Likewise, if CW_CPLUSPLUS_DIR is set, the Studio C++
190 compiler in \fI$CW_CPLUSPLUS_DIR\fp will be used.
191 Otherwise, if SPRO_VROOT is set, the Studio C and C++ compilers
192 in \fI$SPRO_VROOT/bin\fp will be used.
193 Otherwise, if SPRO_ROOT is set, the Studio C and C++ compilers
194 in \fI$SPRO_ROOT/SS12/bin\fp will be used.
195 Otherwise, if BUILD_TOOLS is set, the Studio C and C++ compilers
196 in \fI$BUILD_TOOLS/SUNW$pro/SS12/bin\fp will be used.
197 Otherwise, the Studio compilers in a predefined default location
198 will be used.
199 .TP 4
200 .B CW_GCC, CW_GPLUSPLUS
201 If these variables are set in the environment, they specify the full
202 pathname for the GNU C and C++ compilers, respectively.
203 .TP 4
204 .B CW_GCC_DIR, CW_GPLUSPLUS_DIR, GNU_ROOT
205 If CW_GCC or CW_GPLUSPLUS are not set,
206 these variables alter the search path for GNU compilers in a manner
207 similar to that described above for the Studio compilers. Specifically:
208 .B
209 If CW_GCC_DIR is set, the GNU C compiler in \fI$CW_GCC_DIR\fp
210 will be used. Likewise, if CW_GPLUSPLUS_DIR is set, the GNU C++
211 compiler in \fI$CW_GPLUSPLUS_DIR\fp will be used.
212 Otherwise, if GNU_ROOT is set, the GNU C and C++ compilers
213 in \fI$GCC_ROOT/bin\fp will be used.
214 Otherwise, the GNU compilers in a predefined default location
215 will be used.
216 .SH EXIT STATUS
217 .LP
218 The following exit status values are returned:
219 .Bl -tag -width indent

```

```
202 .It 0
203 The primary compiler, and shadow compilers if invoked, both completed
215 .IP "\fB0\fR" 4
216 The primary compiler, and shadow compiler if invoked, both completed
204 successfully.
205 .It >0
218 .IP "\fB>0\fR" 4
206 A usage error occurred, or one or more compilers returned a nonzero
207 exit status.
208 .El
209 .Sh SEE ALSO
210 .Xr cc 1 ,
211 .Xr CC 1 ,
212 .Xr gcc 1
213 .Sh BUGS
214 The translations provided for gcc are not always exact and in some cases
215 reflect local policy rather than actual equivalence.
216 .Pp
221 .SH BUGS
222 .LP
223 The translations provided for gcc are not always exact and in some
224 cases reflect local ON policy rather than actual equivalence.
225 .LP
217 Additional compiler types should be supported.
218 .Pp
227 .LP
219 The translation engine is hacky.
229 .SH SEE ALSO
230 .LP
231 cc(1), CC(1), gcc(1)
```

```

*****
45601 Thu Feb 22 01:20:20 2018
new/usr/src/tools/cw/cw.c
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****

```

```

2 #endif /* ! codereview */
3 /*
4  * CDDL HEADER START
5  *
6  * The contents of this file are subject to the terms of the
7  * Common Development and Distribution License (the "License").
8  * You may not use this file except in compliance with the License.
9  *
10 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 * or http://www.opensolaris.org/os/licensing.
12 * See the License for the specific language governing permissions
13 * and limitations under the License.
14 *
15 * When distributing Covered Code, include this CDDL HEADER in each
16 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 * If applicable, add the following below this CDDL HEADER, with the
18 * fields enclosed by brackets "[]" replaced with your own identifying
19 * information: Portions Copyright [yyyy] [name of copyright owner]
20 *
21 * CDDL HEADER END
22 */

24 /*
25  * Copyright 2018, Richard Lowe.
26  * Copyright 2011, Richard Lowe.
27 */
28 * Copyright 2010 Sun Microsystems, Inc. All rights reserved.
29 * Use is subject to license terms.
30 */

32 /*
33  * Wrapper for the GNU C compiler to make it accept the Sun C compiler
34  * arguments where possible.
35  *
36  * Since the translation is inexact, this is something of a work-in-progress.
37  *
38  */

40 /* If you modify this file, you must increment CW_VERSION */
41 #define CW_VERSION      "2.0"
42 #define CW_VERSION      "1.30"

43 /*
44  * -#          Verbose mode
45  * -###       Show compiler commands built by driver, no compilation
46  * -A<name{tokens}> Preprocessor predicate assertion
47  * -B<[static|dynamic]> Specify dynamic or static binding
48  * -C         Prevent preprocessor from removing comments
49  * -c         Compile only - produce .o files, suppress linking
50  * -cg92     Alias for -xtarget=ss1000
51  * -D<name[=token]> Associate name with token as if by #define
52  * -d[y|n]   dynamic [-dy] or static [-dn] option to linker

```

```

53 * -E          Compile source through preprocessor only, output to stdout
54 * -erroff=<t> Suppress warnings specified by tags t(%none, %all, <tag list>)
55 * -errtags=<a> Display messages with tags a(no, yes)
56 * -errwarn=<t> Treats warnings specified by tags t(%none, %all, <tag list>)
57 *           as errors
58 * -fast      Optimize using a selection of options
59 * -fd        Report old-style function definitions and declarations
60 * -features=zla Allow zero-length arrays
61 * -flags     Show this summary of compiler options
62 * -fnonstd   Initialize floating-point hardware to non-standard preferences
63 * -fns[=<yes|no>] Select non-standard floating point mode
64 * -fprecision=<p> Set FP rounding precision mode p(single, double, extended)
65 * -fround=<r> Select the IEEE rounding mode in effect at startup
66 * -fsimple[=<n>] Select floating-point optimization preferences <n>
67 * -fsingle   Use single-precision arithmetic (-Xt and -Xs modes only)
68 * -ftrap=<t> Select floating-point trapping mode in effect at startup
69 * -fstore    force floating pt. values to target precision on assignment
70 * -G         Build a dynamic shared library
71 * -g         Compile for debugging
72 * -H         Print path name of each file included during compilation
73 * -h <name> Assign <name> to generated dynamic shared library
74 * -I<dir>    Add <dir> to preprocessor #include file search path
75 * -i         Passed to linker to ignore any LD_LIBRARY_PATH setting
76 * -keeptmp  Keep temporary files created during compilation
77 * -KPIC     Compile position independent code with 32-bit addresses
78 * -Kpic     Compile position independent code
79 * -L<dir>    Pass to linker to add <dir> to the library search path
80 * -l<name>   Link with library lib<name>.a or lib<name>.so
81 * -mc       Remove duplicate strings from .comment section of output files
82 * -mr       Remove all strings from .comment section of output files
83 * -mr,"string" Remove all strings and append "string" to .comment section
84 * -mt       Specify options needed when compiling multi-threaded code
85 * -native   Find available processor, generate code accordingly
86 * -nofstore Do not force floating pt. values to target precision
87 *           on assignment
88 * -nolib    Same as -xnolib
89 * -noqueue  Disable queuing of compiler license requests
90 * -norunpath Do not build in a runtime path for shared libraries
91 * -O        Use default optimization level (-xO2 or -xO3. Check man page.)
92 * -o <outputfile> Set name of output file to <outputfile>
93 * -p        Compile source through preprocessor only, output to .i file
94 * -PIC      Alias for -KPIC or -xcode=pic32
95 * -p        Compile for profiling with prof
96 * -pic      Alias for -Kpic or -xcode=pic13
97 * -Q[y|n]   Emit/don't emit identification info to output file
98 * -qp       Compile for profiling with prof
99 * -R<dir[:dir]> Build runtime search path list into executable
100 * -S        Compile and only generate assembly code (.s)
101 * -s        Strip symbol table from the executable file
102 * -t        Turn off duplicate symbol warnings when linking
103 * -U<name>  Delete initial definition of preprocessor symbol <name>
104 * -V        Report version number of each compilation phase
105 * -v        Do stricter semantic checking
106 * -W<c>,<arg> Pass <arg> to specified component <c> (a,l,m,p,O,2,h,i,u)
107 * -w        Suppress compiler warning messages
108 * -Xa       Compile assuming ANSI C conformance, allow K & R extensions
109 *           (default mode)
110 * -Xc       Compile assuming strict ANSI C conformance
111 * -Xs       Compile assuming (pre-ANSI) K & R C style code
112 * -Xt       Compile assuming K & R conformance, allow ANSI C
113 * -x386     Generate code for the 80386 processor
114 * -x486     Generate code for the 80486 processor
115 * -xarch=<a> Specify target architecture instruction set
116 * -xbuiltin[=<b>] When profitable inline, or substitute intrinsic functions
117 *           for system functions, b={%all,%none}
118 * -xCC      Accept C++ style comments

```

```

119 * -xchar_byte_order=<o> Specify multi-char byte order <o> (default, high, low)
120 * -xchip=<c> Specify the target processor for use by the optimizer
121 * -xcode=<c> Generate different code for forming addresses
122 * -xcrossfile[=<n>] Enable optimization and inlining across source files,
123 * n={0|1}
124 * -xe Perform only syntax/semantic checking, no code generation
125 * -xF Compile for later mapfile reordering or unused section
126 * elimination
127 * -xhelp=<f> Display on-line help information f(flags, readme, errors)
128 * -xildoff Cancel -xildon
129 * -xildon Enable use of the incremental linker, ild
130 * -xinline=[<a>, ..., <a>] Attempt inlining of specified user routines,
131 * <a>={%auto,func,no%func}
132 * -xlibmieee Force IEEE 754 return values for math routines in
133 * exceptional cases
134 * -xlibmil Inline selected libm math routines for optimization
135 * -xlic_lib=sunperf Link in the Sun supplied performance libraries
136 * -xlicinfo Show license server information
137 * -xM Generate makefile dependencies
138 * -xM1 Generate makefile dependencies, but exclude /usr/include
139 * -xmaxopt=[off,1,2,3,4,5] maximum optimization level allowed on #pragma opt
140 * -xnolib Do not link with default system libraries
141 * -xnolibmil Cancel -xlibmil on command line
142 * -xO<n> Generate optimized code (n={1|2|3|4|5})
143 * -xP Print prototypes for function definitions
144 * -xpentium Generate code for the pentium processor
145 * -xpg Compile for profiling with gprof
146 * -xprofile=<p> Collect data for a profile or use a profile to optimize
147 * <p>={{collect,use}[:<path>],tcov}
148 * -xregs=<r> Control register allocation
149 * -xs Allow debugging without object (.o) files
150 * -xsb Compile for use with the WorkShop source browser
151 * -xsbfast Generate only WorkShop source browser info, no compilation
152 * -xsfpconst Represent unsuffixed floating point constants as single
153 * precision
154 * -xspace Do not do optimizations that increase code size
155 * -xstrconst Place string literals into read-only data segment
156 * -xtarget=<t> Specify target system for optimization
157 * -xtmp=<dir> Set directory for temporary files to <dir>
158 * -xtime Report the execution time for each compilation phase
159 * -xtransition Emit warnings for differences between K&R C and ANSI C
160 * -xtrigraphs[=<yes|no>] Enable|disable trigraph translation
161 * -xunroll=n Enable unrolling loops n times where possible
162 * -Y<c>,<dir> Specify <dir> for location of component <c> (a,l,m,p,0,h,i,u)
163 * -YA,<dir> Change default directory searched for components
164 * -YI,<dir> Change default directory searched for include files
165 * -YP,<dir> Change default directory for finding libraries files
166 * -YS,<dir> Change default directory for startup object files
167 */

169 /*
170 * Translation table:
171 */
172 /*
173 * -# -v
174 * -### error
175 * -A<name[(tokens)]> pass-thru
176 * -B<[static|dynamic]> pass-thru (syntax error for anything else)
177 * -C pass-thru
178 * -c pass-thru
179 * -cg92 -m32 -mcpu=v8 -mtune=supersparc (SPARC only)
180 * -D<name[=token]> pass-thru
181 * -dy or -dn -Wl,-dy or -Wl,-dn
182 * -E pass-thru
183 * -erroff=E_EMPTY_TRANSLATION_UNIT ignore
184 * -errtags=%all -Wall

```

```

185 * -errwarn=%all -Werror else -Wno-error
186 * -fast error
187 * -fd error
188 * -features=zla ignore
189 * -flags --help
190 * -fnonstd error
191 * -fns[=<yes|no>] error
192 * -fprecision=<p> error
193 * -fround=<r> error
194 * -fsimple[=<n>] error
195 * -fsingle[=<n>] error
196 * -ftrap=<t> error
197 * -fstore error
198 * -G pass-thru
199 * -g pass-thru
200 * -H pass-thru
201 * -h <name> pass-thru
202 * -I<dir> pass-thru
203 * -i pass-thru
204 * -keeptmp -save-temps
205 * -KPIC -fPIC
206 * -Kpic -fpic
207 * -L<dir> pass-thru
208 * -l<name> pass-thru
209 * -mc error
210 * -mr error
211 * -mr,"string" error
212 * -mt -D_REENTRANT
213 * -native error
214 * -nofstore error
215 * -nolib -nodefaultlibs
216 * -noqueue ignore
217 * -norunpath ignore
218 * -O -O1 (Check the man page to be certain)
219 * -o <outputfile> pass-thru
220 * -P -E -o filename.i (or error)
221 * -PIC -fPIC (C++ only)
222 * -p pass-thru
223 * -pic -fpic (C++ only)
224 * -Q[y|n] error
225 * -qp -p
226 * -R<dir[:dir]> pass-thru
227 * -S pass-thru
228 * -s -Wl,-s
229 * -t -Wl,-t
230 * -U<name> pass-thru
231 * -V --version
232 * -v -Wall
233 * -Wa,<arg> pass-thru
234 * -Wp,<arg> pass-thru except -xc99=<a>
235 * -Wl,<arg> pass-thru
236 * -W{m,0,2,h,i,u} error/ignore
237 * -Wu,-xmodel=kernel -ffreestanding -mcmmodel=kernel -mno-red-zone
238 * -xmodel=kernel -ffreestanding -mcmmodel=kernel -mno-red-zone
239 * -Wu,-save_args -msave_args
240 * -w pass-thru
241 * -Xa -std=iso9899:199409 or -ansi
242 * -Xc -ansi -pedantic
243 * -Xt error
244 * -Xs -traditional -std=c89
245 * -x386 -march=i386 (x86 only)
246 * -x486 -march=i486 (x86 only)
247 * -xarch=<a> table
248 * -xbuiltin[=<b>] -fbuiltin (-fno-builtin otherwise)
249 * -xCC ignore
250 * -xchar_byte_order=<o> error

```

```

251 * -xchip=<c>          table
252 * -xcode=<c>          table
253 * -xdebugformat=<format> ignore (always use dwarf-2 for gcc)
254 * -xcrossfile[=<n>]  ignore
255 * -xe                error
256 * -xF               error
257 * -xhelp=<f>        error
258 * -xildoff          ignore
259 * -xildon           ignore
260 * -xinline          ignore
261 * -xlibmieee       error
262 * -xlibmil         error
263 * -xlic_lib=sunperf error
264 * -xM              -M
265 * -xM1             -MM
266 * -xmaxopt=[...]   error
267 * -xnolib          -nodefaultlibs
268 * -xnolibmil       error
269 * -xO<n>           -O<n>
270 * -xP              error
271 * -xpentium        -march=pentium (x86 only)
272 * -xpg             error
273 * -xprofile=<p>    error
274 * -xregs=<r>       table
275 * -xs              error
276 * -xsb             error
277 * -xsbfast         error
278 * -xsfpconst      error
279 * -xspace          ignore (-not -Os)
280 * -xstrconst       ignore
281 * -xtarget=<t>     table
282 * -xtemp=<dir>    error
283 * -xtime           error
284 * -xtransition     -Wtransition
285 * -xtrigraphs=<yes|no> -trigraphs -notrigraphs
286 * -xunroll=n      error
287 * -W0,-xdbggen=no%usedonly -fno-eliminate-unused-debug-symbols
288 *                  -fno-eliminate-unused-debug-types
289 * -Y<c>,<dir>      error
290 * -YA,<dir>        error
291 * -YI,<dir>        -nostdinc -I<dir>
292 * -YP,<dir>        error
293 * -YS,<dir>        error
294 */

```

```

296 #include <ctype.h>
297 #include <err.h>
298 #include <errno.h>
299 #include <fcntl.h>
300 #include <getopt.h>
301 #endif /* ! codereview */
302 #include <stdio.h>
303 #include <stdlib.h>
304 #include <string.h>
305 #include <unistd.h>

307 #include <sys/param.h>
308 #include <sys/stat.h>
309 #endif /* ! codereview */
310 #include <sys/types.h>
311 #include <unistd.h>
312 #include <string.h>
313 #include <stdlib.h>
314 #include <ctype.h>
315 #include <fcntl.h>
316 #include <errno.h>

```

```

278 #include <stdarg.h>
311 #include <sys/utsname.h>
280 #include <sys/param.h>
281 #include <sys/isa_defs.h>
312 #include <sys/wait.h>
283 #include <sys/stat.h>

314 #define CW_F_CXX      0x01
315 #define CW_F_SHADOW  0x02
316 #define CW_F_EXEC    0x04
317 #define CW_F_ECHO    0x08
318 #define CW_F_XLATE   0x10
319 #define CW_F_PROG    0x20

292 typedef enum cw_compiler {
293     CW_C_CC = 0,
294     CW_C_GCC
295 } cw_compiler_t;

297 static const char *cmds[] = {
298     "cc", "CC",
299     "gcc", "g++"
300 };

302 static char default_dir[2][MAXPATHLEN] = {
303     DEFAULT_CC_DIR,
304     DEFAULT_GCC_DIR,
305 };

307 #define CC(ctx) \
308     (((ctx)->i_flags & CW_F_SHADOW) ? \
309     ((ctx)->i_compiler == CW_C_CC ? CW_C_GCC : CW_C_CC) : \
310     (ctx)->i_compiler)

312 #define CIDX(compiler, flags) \
313     ((int)(compiler) < 1) + ((flags) & CW_F_CXX ? 1 : 0)

321 typedef enum cw_op {
322     CW_O_NONE = 0,
323     CW_O_PREPROCESS,
324     CW_O_COMPILE,
325     CW_O_LINK
326 } cw_op_t;
    unchanged_portion_omitted_

336 typedef enum {
337     GNU,
338     SUN
339 } compiler_style_t;

341 typedef struct {
342     char *c_name;
343     char *c_path;
344     compiler_style_t c_style;
345 } cw_compiler_t;

347 #endif /* ! codereview */
348 typedef struct cw_ictx {
349     cw_compiler_t *i_compiler;
350     cw_compiler_t i_compiler;
351     struct aelist *i_ae;
352     uint32_t i_flags;
353     int i_oldargc;
354     char **i_oldargv;
355     pid_t i_pid;
356     char i_discard[MAXPATHLEN];

```

```

356     char      *i_stderr;
357 } cw_ictx_t;
    unchanged portion omitted

382 /*
383  * The translation table for the -xarch= flag used in the Studio compilers.
384  */
385 static const xarch_table_t xtbl[] = {
386 #if defined(__x86)
387     { "generic",    SS11, {NULL} },
388     { "generic",    SS11 },
389     { "generic64", (SS11|M64), { "-m64", "-mtune=opteron" } },
390     { "amd64",     (SS11|M64), { "-m64", "-mtune=opteron" } },
391     { "386",       (SS11), { "-march=i386" } },
392     { "pentium_pro", SS11, { "-march=pentiumpro" } },
393     { "sse",       (SS11), { "-msse", "-mfpmath=sse" } },
394     { "sse2",      (SS11), { "-msse2", "-mfpmath=sse" } },
395 #elif defined(__sparc)
396     { "generic",    (SS11|M32), { "-m32", "-mcpu=v8" } },
397     { "generic64", (SS11|M64), { "-m64", "-mcpu=v9" } },
398     { "v8",        (SS11|M32), { "-m32", "-mcpu=v8", "-mno-v8plus" } },
399     { "v8plus",    (SS11|M32), { "-m32", "-mcpu=v9", "-mv8plus" } },
400     { "v8plusa",   (SS11|M32), { "-m32", "-mcpu=ultrasparc", "-mv8plus",
401     { "v8plusb",   (SS11|M32), { "-m32", "-mcpu=ultrasparc3", "-mv8plus",
402     { "v9",        (SS11|M64), { "-m64", "-mcpu=v9" } },
403     { "v9a",       (SS11|M64), { "-m64", "-mcpu=ultrasparc", "-mvis" } },
404     { "v9b",       (SS11|M64), { "-m64", "-mcpu=ultrasparc3", "-mvis" } },
405     { "sparc",     (SS12), { "-mcpu=v9", "-mv8plus" } },
406     { "sparcv8",   (SS12), { "-mcpu=ultrasparc", "-mvis" } },
407     { "sparcv8",   (SS12), { "-mcpu=ultrasparc3", "-mvis" } },
408     { "sparcv8",   (SS12), { "-mcpu=ultrasparc3", "-mvis" } },
409 #endif
410 };

412 static int xtbl_size = sizeof (xtbl) / sizeof (xarch_table_t);

395 static const char *progrname;

414 static const char *xchip_tbl[] = {
415 #if defined(__x86)
416     "386",          "-mtune=i386", NULL,
417     "486",          "-mtune=i486", NULL,
418     "pentium",      "-mtune=pentium", NULL,
419     "pentium_pro", "-mtune=pentiumpro", NULL,
420 #elif defined(__sparc)
421     "super",        "-mtune=supersparc", NULL,
422     "ultra",         "-mtune=ultrasparc", NULL,
423     "ultra3",       "-mtune=ultrasparc3", NULL,
424 #endif
425     NULL,           NULL
426 };
    unchanged portion omitted

456 static void
457 nomem(void)
458 {
459     errx(1, "out of memory");
460     (void) fprintf(stderr, "%s: error: out of memory\n", progrname);
461     exit(1);
462 }

463 static void
464 cw_perror(const char *fmt, ...)
465 {
466     va_list ap;

```

```

450     int saved_errno = errno;

452     (void) fprintf(stderr, "%s: error: ", progrname);

454     va_start(ap, fmt);
455     (void) vfprintf(stderr, fmt, ap);
456     va_end(ap);

458     (void) fprintf(stderr, " (%s)\n", strerror(saved_errno));
459 }
    unchanged portion omitted

491 static void
492 error(const char *arg)
493 {
494     errx(2, "error: mapping failed at or near arg '%s'\n", arg);
495     (void) fprintf(stderr, "%s: error: mapping failed at or near arg '%s'\n", progrname, arg);
496     exit(2);
497 }
    unchanged portion omitted

526 /* ARGUSED */
527 static void
528 Xamode(struct aelist __unused *h)
529 Xamode(struct aelist *h)
530 {
531 }
    unchanged portion omitted

556 static void
557 usage()
558 {
559     extern char *progrname;
560 #endif /* ! codereview */
561     (void) fprintf(stderr,
562     "usage: %s [-C] [--versions] --primary <compiler> "
563     "[--shadow <compiler>]... -- cflags...\n",
564     progrname);
565     (void) fprintf(stderr, "compilers take the form: name,path,style\n"
566     " - name: a unique name usable in flag specifiers\n"
567     " - path: path to the compiler binary\n"
568     " - style: the style of flags expected: either sun or gnu\n");
569     (void) fprintf(stderr, "usage: %s { -_cc | -_gcc | -_CC | -_g++ } [ -_compiler | ... ]\n",
570     progrname);
571     exit(2);
572 }
    unchanged portion omitted

619 static void
620 do_gcc(cw_ictx_t *ctx)
621 {
622     int c;
623     int pic = 0, nolIBC = 0;
624     int in_output = 0, seen_o = 0, c_files = 0;
625     cw_op_t op = CW_O_LINK;
626     char *model = NULL;
627     char *nameflag;
628 #endif /* ! codereview */
629     int mflag = 0;

631     if (ctx->i_flags & CW_F_PROG) {
632         newae(ctx->i_ae, "--version");
633         return;
634     }

```

```

636     newae(ctx->i_ae, "-fident");
637     newae(ctx->i_ae, "-finline");
638     newae(ctx->i_ae, "-fno-inline-functions");
639     newae(ctx->i_ae, "-fno-builtin");
640     newae(ctx->i_ae, "-fno-asm");
641     newae(ctx->i_ae, "-fdiagnostics-show-option");
642     newae(ctx->i_ae, "-nodefaultlibs");

644 #if defined(__sparc)
645     /*
646     * The SPARC ldd and std instructions require 8-byte alignment of
647     * their address operand. gcc correctly uses them only when the
648     * ABI requires 8-byte alignment; unfortunately we have a number of
649     * pieces of buggy code that doesn't conform to the ABI. This
650     * flag makes gcc work more like Studio with -xmalign=4.
651     */
652     newae(ctx->i_ae, "-mno-integer-ldd-std");
653 #endif

655     /*
656     * This is needed because 'u' is defined
657     * under a conditional on 'sun'. Should
658     * probably just remove the conditional,
659     * or make it be dependent on '__sun'.
660     *
661     * -Dunix is also missing in enhanced ANSI mode
662     */
663     newae(ctx->i_ae, "-D_sun");

665     if (asprintf(&nameflag, "-_s=", ctx->i_compiler->c_name) == -1)
666         nomem();

668 #endif /* ! codereview */
669     /*
670     * Walk the argument list, translating as we go ..
671     */

672     while (--ctx->i_oldargc > 0) {
673         char *arg = ++ctx->i_oldargv;
674         size_t arglen = strlen(arg);

676         if (*arg == '-') {
677             arglen--;
678         } else {
679             /*
680             * Discard inline files that gcc doesn't grok
681             */
682             if (!in_output && arglen > 3 &&
683                 strcmp(arg + arglen - 3, ".il") == 0)
684                 continue;

686             if (!in_output && arglen > 2 &&
687                 arg[arglen - 2] == '.' &&
688                 (arg[arglen - 1] == 's' || arg[arglen - 1] == 's' ||
689                  arg[arglen - 1] == 'c' || arg[arglen - 1] == 'i'))
690                 c_files++;

692             /*
693             * Otherwise, filenames and partial arguments
694             * are passed through for gcc to chew on. However,
695             * output is always discarded for the secondary
696             * compiler.
697             */
698             if ((ctx->i_flags & CW_F_SHADOW) && in_output)
699                 newae(ctx->i_ae, ctx->i_discard);
700             else

```

```

701                 newae(ctx->i_ae, arg);
702                 in_output = 0;
703                 continue;
704             }

706             if (ctx->i_flags & CW_F_CXX) {
707                 if (strncmp(arg, "-g++=", 6) == 0) {
708                     newae(ctx->i_ae, strchr(arg, '=') + 1);
709                     continue;
710                 }
711 #endif /* ! codereview */
712                 if (strncmp(arg, "-compat=", 8) == 0) {
713                     /* discard -compat=4 and -compat=5 */
714                     continue;
715                 }
716                 if (strcmp(arg, "-Qoption") == 0) {
717                     /* discard -Qoption and its two arguments */
718                     if (ctx->i_oldargc < 3)
719                         error(arg);
720                     ctx->i_oldargc -= 2;
721                     ctx->i_oldargv += 2;
722                     continue;
723                 }
724                 if (strcmp(arg, "-xwe") == 0) {
725                     /* turn warnings into errors */
726                     newae(ctx->i_ae, "-Werror");
727                     continue;
728                 }
729                 if (strcmp(arg, "-noex") == 0) {
730                     /* no exceptions */
731                     newae(ctx->i_ae, "-fno-exceptions");
732                     /* no run time type descriptor information */
733                     newae(ctx->i_ae, "-fno-rtti");
734                     continue;
735                 }
736                 if (strcmp(arg, "-pic") == 0) {
737                     newae(ctx->i_ae, "-fpic");
738                     pic = 1;
739                     continue;
740                 }
741                 if (strcmp(arg, "-PIC") == 0) {
742                     newae(ctx->i_ae, "-fPIC");
743                     pic = 1;
744                     continue;
745                 }
746                 if (strcmp(arg, "-norunpath") == 0) {
747                     /* gcc has no corresponding option */
748                     continue;
749                 }
750                 if (strcmp(arg, "-nolib") == 0) {
751                     /* -nodefaultlibs is on by default */
752                     nolibc = 1;
753                     continue;
754                 }
755 #if defined(__sparc)
756                 if (strcmp(arg, "-cg92") == 0) {
757                     mflag |= xlate_xtb(ctx->i_ae, "v8");
758                     xlate(ctx->i_ae, "super", xchip_tbl);
759                     continue;
760                 }
761 #endif /* __sparc */
762             }

764             switch ((c = arg[1])) {
765                 case '_':
766                     if ((strncmp(arg, nameflag, strlen(nameflag)) == 0) ||

```



```

767     (strcmp(arg, "-gcc=", 6) == 0) ||
768     (strcmp(arg, "-gnu=", 6) == 0) {
769         newae(ctx->i_ae, strchr(arg, '=') + 1);
770     }
771     if (strcmp(arg, "-noecho") == 0)
772         ctx->i_flags &= ~CW_F_ECHO;
773     else if (strcmp(arg, "-cc=", 5) == 0 ||
774             strcmp(arg, "-_CC=", 5) == 0)
775         /* EMPTY */;
776     else if (strcmp(arg, "-gcc=", 6) == 0 ||
777             strcmp(arg, "-_g++=", 6) == 0)
778         newae(ctx->i_ae, arg + 6);
779     else
780         error(arg);
781     break;
782 case '#':
783     if (arglen == 1) {
784         newae(ctx->i_ae, "-v");
785         break;
786     }
787     error(arg);
788     break;
789 case 'g':
790     newae(ctx->i_ae, "-gdwarf-2");
791     break;
792 case 'E':
793     if (arglen == 1) {
794         newae(ctx->i_ae, "-xc");
795         newae(ctx->i_ae, arg);
796         op = CW_O_PREPROCESS;
797         nolibc = 1;
798         break;
799     }
800     error(arg);
801     break;
802 case 'c':
803 case 'S':
804     if (arglen == 1) {
805         op = CW_O_COMPILE;
806         nolibc = 1;
807     }
808     /* FALLTHROUGH */
809 case 'C':
810 case 'H':
811 case 'p':
812     if (arglen == 1) {
813         newae(ctx->i_ae, arg);
814         break;
815     }
816     error(arg);
817     break;
818 case 'A':
819 case 'h':
820 case 'I':
821 case 'i':
822 case 'L':
823 case 'l':
824 case 'R':
825 case 'U':
826 case 'u':
827 case 'w':
828     newae(ctx->i_ae, arg);
829     break;
830 case 'o':
831     seen_o = 1;
832     if (arglen == 1) {

```

```

823         in_output = 1;
824         newae(ctx->i_ae, arg);
825     } else if (ctx->i_flags & CW_F_SHADOW) {
826         newae(ctx->i_ae, "-o");
827         newae(ctx->i_ae, ctx->i_discard);
828     } else {
829         newae(ctx->i_ae, arg);
830     }
831     break;
832 case 'D':
833     newae(ctx->i_ae, arg);
834     /*
835      * XXX Clearly a hack ... do we need _KADB too?
836      */
837     if (strcmp(arg, "-D_KERNEL") == 0 ||
838         strcmp(arg, "-D_BOOT") == 0)
839         newae(ctx->i_ae, "-ffreestanding");
840     break;
841 case 'd':
842     if (arglen == 2) {
843         if (strcmp(arg, "-dy") == 0) {
844             newae(ctx->i_ae, "-WL,-dy");
845             break;
846         }
847         if (strcmp(arg, "-dn") == 0) {
848             newae(ctx->i_ae, "-WL,-dn");
849             break;
850         }
851     }
852     if (strcmp(arg, "-dalign") == 0) {
853         /*
854          * -dalign forces alignment in some cases;
855          * gcc does not need any flag to do this.
856          */
857         break;
858     }
859     error(arg);
860     break;
861 case 'e':
862     if (strcmp(arg,
863             "-erroff=E_EMPTY_TRANSLATION_UNIT") == 0) {
864         /*
865          * Accept but ignore this -- gcc doesn't
866          * seem to complain about empty translation
867          * units
868          */
869         break;
870     }
871     /* XX64 -- ignore all -erroff= options, for now */
872     if (strcmp(arg, "-erroff=", 8) == 0)
873         break;
874     if (strcmp(arg, "-errtags=yes") == 0) {
875         warnings(ctx->i_ae);
876         break;
877     }
878     if (strcmp(arg, "-errwarn=%all") == 0) {
879         newae(ctx->i_ae, "-Werror");
880         break;
881     }
882     error(arg);
883     break;
884 case 'f':
885     if (strcmp(arg, "-flags") == 0) {
886         newae(ctx->i_ae, "--help");
887         break;
888     }

```

```

889     if (strncmp(arg, "--features=zla", 13) == 0) {
890         /*
891          * Accept but ignore this -- gcc allows
892          * zero length arrays.
893          */
894         break;
895     }
896     error(arg);
897     break;
898 case 'G':
899     newae(ctx->i_ae, "--shared");
900     nolibc = 1;
901     break;
902 case 'k':
903     if (strcmp(arg, "-keeptmp") == 0) {
904         newae(ctx->i_ae, "-save-temps");
905         break;
906     }
907     error(arg);
908     break;
909 case 'K':
910     if (arglen == 1) {
911         if ((arg = *++ctx->i_oldargv) == NULL ||
912             *arg == '\\0')
913             error("-K");
914         ctx->i_oldargc--;
915     } else {
916         arg += 2;
917     }
918     if (strcmp(arg, "pic") == 0) {
919         newae(ctx->i_ae, "-fpic");
920         pic = 1;
921         break;
922     }
923     if (strcmp(arg, "PIC") == 0) {
924         newae(ctx->i_ae, "-fPIC");
925         pic = 1;
926         break;
927     }
928     error("-K");
929     break;
930 case 'm':
931     if (strcmp(arg, "-mt") == 0) {
932         newae(ctx->i_ae, "-D_REENTRANT");
933         break;
934     }
935     if (strcmp(arg, "-m64") == 0) {
936         newae(ctx->i_ae, "-m64");
937 #if defined(__x86)
938         newae(ctx->i_ae, "-mtune=opteron");
939 #endif
940         mflag |= M64;
941         break;
942     }
943     if (strcmp(arg, "-m32") == 0) {
944         newae(ctx->i_ae, "-m32");
945         mflag |= M32;
946         break;
947     }
948     error(arg);
949     break;
950 case 'B': /* linker options */
951 case 'M':
952 case 'z':
953     {
954         char *opt;

```

```

955         size_t len;
956         char *s;

958         if (arglen == 1) {
959             opt = *++ctx->i_oldargv;
960             if (opt == NULL || *opt == '\\0')
961                 error(arg);
962             ctx->i_oldargc--;
963         } else {
964             opt = arg + 2;
965         }
966         len = strlen(opt) + 7;
967         if ((s = malloc(len)) == NULL)
968             nomem();
969         (void) snprintf(s, len, "-Wl,-%c%s", c, opt);
970         newae(ctx->i_ae, s);
971         free(s);
972     }
973     break;
974 case 'n':
975     if (strcmp(arg, "-noqueue") == 0) {
976         /*
977          * Horrid license server stuff - n/a
978          */
979         break;
980     }
981     error(arg);
982     break;
983 case 'O':
984     if (arglen == 1) {
985         newae(ctx->i_ae, "-O");
986         break;
987     }
988     error(arg);
989     break;
990 case 'P':
991     /*
992      * We could do '-E -o filename.i', but that's hard,
993      * and we don't need it for the case that's triggering
994      * this addition. We'll require the user to specify
995      * -o in the Makefile. If they don't they'll find out
996      * in a hurry.
997      */
998     newae(ctx->i_ae, "-E");
999     op = CW_O_PREPROCESS;
1000     nolibc = 1;
1001     break;
1002 case 'q':
1003     if (strcmp(arg, "-qp") == 0) {
1004         newae(ctx->i_ae, "-p");
1005         break;
1006     }
1007     error(arg);
1008     break;
1009 case 's':
1010     if (arglen == 1) {
1011         newae(ctx->i_ae, "-Wl,-s");
1012         break;
1013     }
1014     error(arg);
1015     break;
1016 case 't':
1017     if (arglen == 1) {
1018         newae(ctx->i_ae, "-Wl,-t");
1019         break;
1020     }

```

```

1021     error(arg);
1022     break;
1023 case 'V':
1024     if (arglen == 1) {
1025         ctx->i_flags &= ~CW_F_ECHO;
1026         newae(ctx->i_ae, "--version");
1027         break;
1028     }
1029     error(arg);
1030     break;
1031 case 'v':
1032     if (arglen == 1) {
1033         warnings(ctx->i_ae);
1034         break;
1035     }
1036     error(arg);
1037     break;
1038 case 'W':
1039     if (strncmp(arg, "-Wp,-xc99", 9) == 0) {
1040         /*
1041          * gcc's preprocessor will accept c99
1042          * regardless, so accept and ignore.
1043          */
1044         break;
1045     }
1046     if (strncmp(arg, "-Wa,", 4) == 0 ||
1047         strncmp(arg, "-Wp,", 4) == 0 ||
1048         strncmp(arg, "-Wl,", 4) == 0) {
1049         newae(ctx->i_ae, arg);
1050         break;
1051     }
1052     if (strcmp(arg, "-W0,-xc99=pragma") == 0) {
1053         /* (undocumented) enables _Pragma */
1054         break;
1055     }
1056     if (strcmp(arg, "-W0,-xc99=%none") == 0) {
1057         /*
1058          * This is a polite way of saying
1059          * "no c99 constructs allowed!"
1060          * For now, just accept and ignore this.
1061          */
1062         break;
1063     }
1064     if (strcmp(arg, "-W0,-noglobal") == 0 ||
1065         strcmp(arg, "-W0,-xglobalstatic") == 0) {
1066         /*
1067          * gcc doesn't prefix local symbols
1068          * in debug mode, so this is not needed.
1069          */
1070         break;
1071     }
1072     if (strcmp(arg, "-W0,-Lt") == 0) {
1073         /*
1074          * Generate tests at the top of loops.
1075          * There is no direct gcc equivalent, ignore.
1076          */
1077         break;
1078     }
1079     if (strcmp(arg, "-W0,-xdbggen=no%usedonly") == 0) {
1080         newae(ctx->i_ae,
1081             "-fno-eliminate-unused-debug-symbols");
1082         newae(ctx->i_ae,
1083             "-fno-eliminate-unused-debug-types");
1084         break;
1085     }
1086     if (strcmp(arg, "-W2,-xwrap_int") == 0) {

```

```

1087         /*
1088          * Use the legacy behaviour (pre-SS11)
1089          * for integer wrapping.
1090          * gcc does not need this.
1091          */
1092         break;
1093     }
1094     if (strcmp(arg, "-W2,-Rcond_elim") == 0) {
1095         /*
1096          * Elimination and expansion of conditionals;
1097          * gcc has no direct equivalent.
1098          */
1099         break;
1100     }
1101     if (strcmp(arg, "-Wd,-xsafe=unboundsym") == 0) {
1102         /*
1103          * Prevents optimizing away checks for
1104          * unbound weak symbol addresses. gcc does
1105          * not do this, so it's not needed.
1106          */
1107         break;
1108     }
1109     if (strncmp(arg, "-Wc,-xcode=", 11) == 0) {
1110         xlate(ctx->i_ae, arg + 11, xcode_tbl);
1111         if (strncmp(arg + 11, "pic", 3) == 0)
1112             pic = 1;
1113         break;
1114     }
1115     if (strncmp(arg, "-Wc,-Qiselect", 13) == 0) {
1116         /*
1117          * Prevents insertion of register symbols.
1118          * gcc doesn't do this, so ignore it.
1119          */
1120         break;
1121     }
1122     if (strcmp(arg, "-Wc,-Qassembler-ounrefsym=0") == 0) {
1123         /*
1124          * Prevents optimizing away of static variables.
1125          * gcc does not do this, so it's not needed.
1126          */
1127         break;
1128     }
1129     #if defined(__x86)
1130     if (strcmp(arg, "-Wu,-xmodel=kernel") == 0) {
1131         newae(ctx->i_ae, "-ffreestanding");
1132         newae(ctx->i_ae, "-mno-red-zone");
1133         model = "-mcmmodel=kernel";
1134         nolibc = 1;
1135         break;
1136     }
1137     if (strcmp(arg, "-Wu,-save_args") == 0) {
1138         newae(ctx->i_ae, "-msave_args");
1139         break;
1140     }
1141     #endif /* __x86 */
1142     error(arg);
1143     break;
1144 case 'X':
1145     if (strcmp(arg, "-Xa") == 0 ||
1146         strcmp(arg, "-Xt") == 0) {
1147         Xamode(ctx->i_ae);
1148         break;
1149     }
1150     if (strcmp(arg, "-Xc") == 0) {
1151         Xcmode(ctx->i_ae);
1152         break;

```

```

1153     }
1154     if (strcmp(arg, "-Xs") == 0) {
1155         Xsmode(ctx->i_ae);
1156         break;
1157     }
1158     error(arg);
1159     break;
1160 case 'x':
1161     if (arglen == 1)
1162         error(arg);
1163     switch (arg[2]) {
1164 #if defined(__x86)
1165     case '3':
1166         if (strcmp(arg, "-x386") == 0) {
1167             newae(ctx->i_ae, "-march=i386");
1168             break;
1169         }
1170         error(arg);
1171         break;
1172     case '4':
1173         if (strcmp(arg, "-x486") == 0) {
1174             newae(ctx->i_ae, "-march=i486");
1175             break;
1176         }
1177         error(arg);
1178         break;
1179 #endif /* __x86 */
1180     case 'a':
1181         if (strcmp(arg, "-xarch=", 7) == 0) {
1182             mflag |= xlate_xtb(ctx->i_ae, arg + 7);
1183             break;
1184         }
1185         error(arg);
1186         break;
1187     case 'b':
1188         if (strcmp(arg, "-xbuiltin=", 10) == 0) {
1189             if (strcmp(arg + 10, "%all")
1190                 newae(ctx->i_ae, "-fbuiltin");
1191             break;
1192         }
1193         error(arg);
1194         break;
1195     case 'C':
1196         /* Accept C++ style comments -- ignore */
1197         if (strcmp(arg, "-xCC") == 0)
1198             break;
1199         error(arg);
1200         break;
1201     case 'c':
1202         if (strcmp(arg, "-xc99=%all", 10) == 0) {
1203             newae(ctx->i_ae, "-std=gnu99");
1204             break;
1205         }
1206         if (strcmp(arg, "-xc99=%none", 11) == 0) {
1207             newae(ctx->i_ae, "-std=gnu89");
1208             break;
1209         }
1210         if (strcmp(arg, "-xchip=", 7) == 0) {
1211             xlate(ctx->i_ae, arg + 7, xchip_tbl);
1212             break;
1213         }
1214         if (strcmp(arg, "-xcode=", 7) == 0) {
1215             xlate(ctx->i_ae, arg + 7, xcode_tbl);
1216             if (strcmp(arg + 7, "pic", 3) == 0)
1217                 pic = 1;
1218             break;

```

```

1219     }
1220     if (strcmp(arg, "-xcache=", 8) == 0)
1221         break;
1222     if (strcmp(arg, "-xcrossfile", 11) == 0)
1223         break;
1224     error(arg);
1225     break;
1226 case 'd':
1227     if (strcmp(arg, "-xdepend") == 0)
1228         break;
1229     if (strcmp(arg, "-xdebugformat=", 14) == 0)
1230         break;
1231     error(arg);
1232     break;
1233 case 'F':
1234     /*
1235      * Compile for mapfile reordering, or unused
1236      * section elimination, syntax can be -xF or
1237      * more complex, like -xF=%all -- ignore.
1238      */
1239     if (strcmp(arg, "-xF", 3) == 0)
1240         break;
1241     error(arg);
1242     break;
1243 case 'i':
1244     if (strcmp(arg, "-xinline", 8) == 0)
1245         /* No inlining; ignore */
1246         break;
1247     if (strcmp(arg, "-xildon") == 0 ||
1248         strcmp(arg, "-xildoff") == 0)
1249         /* No incremental linking; ignore */
1250         break;
1251     error(arg);
1252     break;
1253 #if defined(__x86)
1254 case 'm':
1255     if (strcmp(arg, "-xmodel=kernel") == 0) {
1256         newae(ctx->i_ae, "-ffreestanding");
1257         newae(ctx->i_ae, "-mno-red-zone");
1258         model = "-mcmmodel=kernel";
1259         nolibc = 1;
1260         break;
1261     }
1262     error(arg);
1263     break;
1264 #endif /* __x86 */
1265 case 'M':
1266     if (strcmp(arg, "-xM") == 0) {
1267         newae(ctx->i_ae, "-M");
1268         break;
1269     }
1270     if (strcmp(arg, "-xM1") == 0) {
1271         newae(ctx->i_ae, "-MM");
1272         break;
1273     }
1274     error(arg);
1275     break;
1276 case 'n':
1277     if (strcmp(arg, "-xnolib") == 0) {
1278         nolibc = 1;
1279         break;
1280     }
1281     error(arg);
1282     break;
1283 case 'O':
1284     if (strcmp(arg, "-xO", 3) == 0) {

```

```

1285     size_t len = strlen(arg);
1286     char *s;
1287     int c = *(arg + 3);
1288     int level;

1290     if (len != 4 || !isdigit(c))
1291         error(arg);

1293     if ((s = malloc(len)) == NULL)
1294         nomem();

1296     level = atoi(arg + 3);
1297     if (level > 5)
1298         error(arg);
1299     if (level >= 2) {
1300         /*
1301          * For gcc-3.4.x at -O2 we
1302          * need to disable optimizations
1303          * that break ON.
1304          */
1305         optim_disable(ctx->i_ae, level);
1306         /*
1307          * limit -xO3 to -O2 as well.
1308          */
1309         level = 2;
1310     }
1311     (void) snprintf(s, len, "-O%d", level);
1312     newae(ctx->i_ae, s);
1313     free(s);
1314     break;
1315 }
1316 error(arg);
1317 break;
1318 case 'p':
1319     if (strcmp(arg, "-xpentium") == 0) {
1320         newae(ctx->i_ae, "-march=pentium");
1321         break;
1322     }
1323     if (strcmp(arg, "-xpg") == 0) {
1324         newae(ctx->i_ae, "-pg");
1325         break;
1326     }
1327     error(arg);
1328     break;
1329 case 'r':
1330     if (strncmp(arg, "-xregs=", 7) == 0) {
1331         xlate(ctx->i_ae, arg + 7, xregs_tbl);
1332         break;
1333     }
1334     error(arg);
1335     break;
1336 case 's':
1337     if (strcmp(arg, "-xs") == 0 ||
1338         strcmp(arg, "-xspace") == 0 ||
1339         strcmp(arg, "-xstrconst") == 0)
1340         break;
1341     error(arg);
1342     break;
1343 case 't':
1344     if (strcmp(arg, "-xtransition") == 0) {
1345         newae(ctx->i_ae, "-Wtransition");
1346         break;
1347     }
1348     if (strcmp(arg, "-xtrigraphs=yes") == 0) {
1349         newae(ctx->i_ae, "-trigraphs");
1350         break;

```

```

1351     }
1352     if (strcmp(arg, "-xtrigraphs=no") == 0) {
1353         newae(ctx->i_ae, "-notrigraphs");
1354         break;
1355     }
1356     if (strncmp(arg, "-xtarget=", 9) == 0) {
1357         xlate(ctx->i_ae, arg + 9, xtarget_tbl);
1358         break;
1359     }
1360     error(arg);
1361     break;
1362 case 'e':
1363 case 'h':
1364 case 'l':
1365     default:
1366         error(arg);
1367         break;
1368 }
1369 break;
1370 case 'Y':
1371     if (arglen == 1) {
1372         if ((arg = *++ctx->i_oldargv) == NULL ||
1373             *arg == '\0')
1374             error("-Y");
1375         ctx->i_oldargc--;
1376         arglen = strlen(arg + 1);
1377     } else {
1378         arg += 2;
1379     }
1380     /* Just ignore -YS,... for now */
1381     if (strncmp(arg, "S", 2) == 0)
1382         break;
1383     if (strncmp(arg, "l", 2) == 0) {
1384         char *s = strdup(arg);
1385         s[0] = '-';
1386         s[1] = 'B';
1387         newae(ctx->i_ae, s);
1388         free(s);
1389         break;
1390     }
1391     if (strncmp(arg, "I", 2) == 0) {
1392         char *s = strdup(arg);
1393         s[0] = '-';
1394         s[1] = 'I';
1395         newae(ctx->i_ae, "-nostdinc");
1396         newae(ctx->i_ae, s);
1397         free(s);
1398         break;
1399     }
1400     error(arg);
1401     break;
1402 case 'Q':
1403     /*
1404      * We could map -Qy into -Wl,-Qy etc.
1405      */
1406     default:
1407         error(arg);
1408         break;
1409     }
1410 }

1412     free(nameflag);

1414 #endif /* ! codereview */
1415     if (c_files > 1 && (ctx->i_flags & CW_F_SHADOW) &&
1416         op != CW_O_PREPROCESS) {

```

```

1417         errx(2, "multiple source files are "
1418             "allowed only with -E or -P");
1307         (void) fprintf(stderr, "%s: error: multiple source files are "
1308             "allowed only with -E or -P\n", progname);
1309         exit(2);
1419     }

1421     /*
1422     * Make sure that we do not have any unintended interactions between
1423     * the xarch options passed in and the version of the Studio compiler
1424     * used.
1425     */
1426     if ((mflag & (SS11|SS12)) == (SS11|SS12)) {
1427         errx(2,
1318         (void) fprintf(stderr,
1428             "Conflicting \"-xarch=\" flags (both Studio 11 and 12)\n");
1320         exit(2);
1429     }

1431     switch (mflag) {
1432     case 0:
1433         /* FALLTHROUGH */
1434     case M32:
1435     #if defined(__sparc)
1436         /*
1437         * Only -m32 is defined and so put in the missing xarch
1438         * translation.
1439         */
1440         newae(ctx->i_ae, "-mcpu=v8");
1441         newae(ctx->i_ae, "-mno-v8plus");
1442     #endif
1443         break;
1444     case M64:
1445     #if defined(__sparc)
1446         /*
1447         * Only -m64 is defined and so put in the missing xarch
1448         * translation.
1449         */
1450         newae(ctx->i_ae, "-mcpu=v9");
1451     #endif
1452         break;
1453     case SS12:
1454     #if defined(__sparc)
1455         /* no -m32/-m64 flag used - this is an error for sparc builds */
1456         (void) fprintf(stderr, "No -m32/-m64 flag defined\n");
1457         exit(2);
1458     #endif
1459         break;
1460     case SS11:
1461         /* FALLTHROUGH */
1462     case (SS11|M32):
1463     case (SS11|M64):
1464         break;
1465     case (SS12|M32):
1466     #if defined(__sparc)
1467         /*
1468         * Need to add in further 32 bit options because with SS12
1469         * the xarch=sparcvis option can be applied to 32 or 64
1470         * bit, and so the translation table (xtbl) cannot handle
1471         * that.
1472         */
1473         newae(ctx->i_ae, "-mv8plus");
1474     #endif
1475         break;
1476     case (SS12|M64):
1477         break;

```

```

1478     default:
1479         (void) fprintf(stderr,
1480             "Incompatible -xarch= and/or -m32/-m64 options used.\n");
1481         exit(2);
1482     }

1484     if ((op == CW_O_LINK || op == CW_O_PREPROCESS) &&
1485         (ctx->i_flags & CW_F_SHADOW))
1375     if (op == CW_O_LINK && (ctx->i_flags & CW_F_SHADOW))
1486         exit(0);

1488     if (model && !pic)
1489         newae(ctx->i_ae, model);
1490     if (!nolibc)
1491         newae(ctx->i_ae, "-lc");
1492     if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1493         newae(ctx->i_ae, "-o");
1494         newae(ctx->i_ae, ctx->i_discard);
1495     }
1496 }

1498 static void
1499 do_cc(cw_ictx_t *ctx)
1500 {
1501     int in_output = 0, seen_o = 0;
1502     cw_op_t op = CW_O_LINK;
1503     char *nameflag;
1504     #ifndef !codereview */
1506     if (ctx->i_flags & CW_F_PROG) {
1507         newae(ctx->i_ae, "-V");
1508         return;
1509     }

1511     if (asprintf(&nameflag, "-_s=", ctx->i_compiler->c_name) == -1)
1512         nomem();

1514     #ifndef !codereview */
1515     while (--ctx->i_oldargc > 0) {
1516         char *arg = ++ctx->i_oldargv;

1518         if (strncmp(arg, "-_cc=", 5) == 0) {
1519             newae(ctx->i_ae, strchr(arg, '=') + 1);
1520             continue;
1521         }

1523     #ifndef !codereview */
1524     if (*arg != '-') {
1525         if (in_output == 0 || !(ctx->i_flags & CW_F_SHADOW)) {
1526             newae(ctx->i_ae, arg);
1527         } else {
1528             in_output = 0;
1529             newae(ctx->i_ae, ctx->i_discard);
1530         }
1531         continue;
1532     }
1533     switch (*(arg + 1)) {
1534     case '_':
1535         if ((strncmp(arg, nameflag, strlen(nameflag)) == 0) ||
1536             (strncmp(arg, "-_cc=", 5) == 0) ||
1537             (strncmp(arg, "-_sun=", 6) == 0)) {
1538             newae(ctx->i_ae, strchr(arg, '=') + 1);
1539         }
1539         if (strcmp(arg, "-_noecho") == 0) {
1539             ctx->i_flags &= ~CW_F_ECHO;
1539         } else if (strncmp(arg, "-_cc=", 5) == 0 ||
1539             strncmp(arg, "-_cc=", 5) == 0) {

```

```

1397     newae(ctx->i_ae, arg + 5);
1398     } else if (strncmp(arg, "-_gcc=", 6) != 0 &&
1399     strncmp(arg, "-_g++=", 6) != 0) {
1400         (void) fprintf(stderr,
1401             "%s: invalid argument '%s'\n", progname,
1402             arg);
1403         exit(2);
1404     }
1405     break;
1406 }
1407
1408 #endif /* ! codereview */
1409 case 'V':
1410     ctx->i_flags &= ~CW_F_ECHO;
1411     newae(ctx->i_ae, arg);
1412     break;
1413 case 'o':
1414     seen_o = 1;
1415     if (strlen(arg) == 2) {
1416         in_output = 1;
1417         newae(ctx->i_ae, arg);
1418     } else if (ctx->i_flags & CW_F_SHADOW) {
1419         newae(ctx->i_ae, "-o");
1420         newae(ctx->i_ae, ctx->i_discard);
1421     } else {
1422         newae(ctx->i_ae, arg);
1423     }
1424     break;
1425 case 'c':
1426 case 'S':
1427     if (strlen(arg) == 2)
1428         op = CW_O_COMPILE;
1429     newae(ctx->i_ae, arg);
1430     break;
1431 case 'E':
1432 case 'P':
1433     if (strlen(arg) == 2)
1434         op = CW_O_PREPROCESS;
1435     /*FALLTHROUGH*/
1436 default:
1437     newae(ctx->i_ae, arg);
1438 }
1439 }
1440
1441 free(nameflag);
1442
1443 #endif /* ! codereview */
1444 if ((op == CW_O_LINK || op == CW_O_PREPROCESS) &&
1445     (ctx->i_flags & CW_F_SHADOW))
1446     exit(0);
1447
1448 if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1449     newae(ctx->i_ae, "-o");
1450     newae(ctx->i_ae, ctx->i_discard);
1451 }
1452 }
1453
1454 static void
1455 prepctx(cw_ictx_t *ctx)
1456 {
1457     newae(ctx->i_ae, ctx->i_compiler->c_path);
1458     const char *dir = NULL, *cmd;
1459     char *program = NULL;
1460     size_t len;
1461
1462     switch (CIDX(CC(ctx), ctx->i_flags)) {
1463     case CIDX(CW_C_CC, 0):

```

```

1412     program = getenv("CW_CC");
1413     dir = getenv("CW_CC_DIR");
1414     break;
1415 case CIDX(CW_C_CC, CW_F_CXX):
1416     program = getenv("CW_CPLUSPLUS");
1417     dir = getenv("CW_CPLUSPLUS_DIR");
1418     break;
1419 case CIDX(CW_C_GCC, 0):
1420     program = getenv("CW_GCC");
1421     dir = getenv("CW_GCC_DIR");
1422     break;
1423 case CIDX(CW_C_GCC, CW_F_CXX):
1424     program = getenv("CW_GPLUSPLUS");
1425     dir = getenv("CW_GPLUSPLUS_DIR");
1426     break;
1427 }
1428
1429 if (program == NULL) {
1430     if (dir == NULL)
1431         dir = default_dir[CC(ctx)];
1432     cmd = cmds[CIDX(CC(ctx), ctx->i_flags)];
1433     len = strlen(dir) + strlen(cmd) + 2;
1434     if ((program = malloc(len)) == NULL)
1435         nomem();
1436     (void) snprintf(program, len, "%s/%s", dir, cmd);
1437 }
1438
1439 newae(ctx->i_ae, program);
1440
1441 if (ctx->i_flags & CW_F_PROG) {
1442     (void) printf("%s: %s\n", (ctx->i_flags & CW_F_SHADOW) ?
1443         "shadow" : "primary", ctx->i_compiler->c_path);
1444     "shadow" : "primary", program);
1445     (void) fflush(stdout);
1446 }
1447
1448 if (!(ctx->i_flags & CW_F_XLATE))
1449     return;
1450
1451 switch (ctx->i_compiler->c_style) {
1452 case SUN:
1453     switch (CC(ctx)) {
1454     case CW_C_CC:
1455         do_cc(ctx);
1456         break;
1457     case GNU:
1458     case CW_C_GCC:
1459         do_gcc(ctx);
1460         break;
1461     }
1462 }
1463
1464 static int
1465 invoke(cw_ictx_t *ctx)
1466 {
1467     char **newargv;
1468     int ac;
1469     struct ae *a;
1470
1471     if ((newargv = calloc(sizeof(*newargv), ctx->i_ae->ael_argc + 1)) ==
1472         NULL)
1473         nomem();
1474
1475     if (ctx->i_flags & CW_F_ECHO)
1476         (void) fprintf(stderr, "+ ");

```

```

1626     for (ac = 0, a = ctx->i_ae->ael_head; a; a = a->ae_next, ac++) {
1627         newargv[ac] = a->ae_arg;
1628         if (ctx->i_flags & CW_F_ECHO)
1629             (void) fprintf(stderr, "%s ", a->ae_arg);
1630         if (a == ctx->i_ae->ael_tail)
1631             break;
1632     }

1634     if (ctx->i_flags & CW_F_ECHO) {
1635         (void) fprintf(stderr, "\n");
1636         (void) fflush(stderr);
1637     }

1639     if (!(ctx->i_flags & CW_F_EXEC))
1640         return (0);

1642     /*
1643     * We must fix up the environment here so that the
1644     * dependency files are not trampled by the shadow compiler.
1645     */
1646     if ((ctx->i_flags & CW_F_SHADOW) &&
1647         (unsetenv("SUNPRO_DEPENDENCIES") != 0 ||
1648          unsetenv("DEPENDENCIES_OUTPUT") != 0)) {
1649         (void) fprintf(stderr, "error: environment setup failed: %s\n",
1650                        strerror(errno));
1651         return (-1);
1652     }

1654     (void) execv(newargv[0], newargv);
1655     warn("couldn't run %s", newargv[0]);
1656     cw_perror("couldn't run %s", newargv[0]);

1657     return (-1);
1658 }

1660 static int
1661 reap(cw_ictx_t *ctx)
1662 {
1663     int status, ret = 0;
1664     char buf[1024];
1665     struct stat s;

1667     /*
1668     * Only wait for one specific child.
1669     */
1670     if (ctx->i_pid <= 0)
1671         return (-1);

1673     do {
1674         if (waitpid(ctx->i_pid, &status, 0) < 0) {
1675             warn("cannot reap child");
1676             cw_perror("cannot reap child");
1677             return (-1);
1678         }
1679         if (status != 0) {
1680             if (WIFSIGNALED(status)) {
1681                 ret = -WTERMSIG(status);
1682                 break;
1683             } else if (WIFEXITED(status)) {
1684                 ret = WEXITSTATUS(status);
1685                 break;
1686             }
1687         } while (!WIFEXITED(status) && !WIFSIGNALED(status));

1689     (void) unlink(ctx->i_discard);

```

```

1691     if (stat(ctx->i_stderr, &s) < 0) {
1692         warn("stat failed on child cleanup");
1693         cw_perror("stat failed on child cleanup");
1694         return (-1);
1695     }
1696     if (s.st_size != 0) {
1697         FILE *f;

1698         if ((f = fopen(ctx->i_stderr, "r")) != NULL) {
1699             while (fgets(buf, sizeof(buf), f))
1700                 (void) fprintf(stderr, "%s", buf);
1701             (void) fflush(stderr);
1702             (void) fclose(f);
1703         }
1704     }
1705     (void) unlink(ctx->i_stderr);
1706     free(ctx->i_stderr);

1708     /*
1709     * cc returns an error code when given -V; we want that to succeed.
1710     */
1711     if (ctx->i_flags & CW_F_PROG)
1712         return (0);

1714     return (ret);
1715 }

1717 static int
1718 exec_ctx(cw_ictx_t *ctx, int block)
1719 {
1720     char *file;

1722     /*
1723     * To avoid offending cc's sensibilities, the name of its output
1724     * file must end in '.o'.
1725     */
1726     if ((file = tmpnam(NULL, ".cw")) == NULL) {
1727         nomem();
1728         return (-1);
1729     }
1730     (void) strcpy(ctx->i_discard, file, MAXPATHLEN);
1731     (void) strcat(ctx->i_discard, ".o", MAXPATHLEN);
1732     free(file);

1734     if ((ctx->i_stderr = tmpnam(NULL, ".cw")) == NULL) {
1735         nomem();
1736         return (-1);
1737     }

1739     if ((ctx->i_pid = fork()) == 0) {
1740         int fd;

1742         (void) fclose(stderr);
1743         if ((fd = open(ctx->i_stderr, O_WRONLY | O_CREAT | O_EXCL,
1744                      0666)) < 0) {
1745             err(1, "open failed for standard error");
1746             cw_perror("open failed for standard error");
1747             exit(1);
1748         }
1749         if (dup2(fd, 2) < 0) {
1750             err(1, "dup2 failed for standard error");
1751             cw_perror("dup2 failed for standard error");
1752             exit(1);
1753         }
1754     }
1755     if (fd != 2)

```



```

1751         (void) close(fd);
1752         if (freopen("/dev/fd/2", "w", stderr) == NULL) {
1753             err(1, "freopen failed for /dev/fd/2");
1754             cw_perror("freopen failed for /dev/fd/2");
1755             exit(1);
1756         }
1757     }
1758     prepctx(ctx);
1759     exit(invoke(ctx));
1760 }
1761
1762 if (ctx->i_pid < 0) {
1763     err(1, "fork failed");
1764     cw_perror("fork failed");
1765     return (1);
1766 }
1767
1768 if (block)
1769     return (reap(ctx));
1770
1771 return (0);
1772 }
1773
1774 static int
1775 parse_compiler(const char *spec, cw_compiler_t *compiler)
1776 {
1777     char *tspec, *token;
1778
1779     if ((tspec = strdup(spec)) == NULL)
1780         nomem();
1781
1782     if ((token = strtok(tspec, ",")) == NULL)
1783         errx(1, "Compiler is missing a name: %s", spec);
1784     compiler->c_name = token;
1785
1786     if ((token = strtok(NULL, ",")) == NULL)
1787         errx(1, "Compiler is missing a path: %s", spec);
1788     compiler->c_path = token;
1789
1790     if ((token = strtok(NULL, ",")) == NULL)
1791         errx(1, "Compiler is missing a style: %s", spec);
1792
1793     if ((strcasecmp(token, "gnu") == 0) ||
1794         (strcasecmp(token, "gcc") == 0))
1795         compiler->c_style = GNU;
1796     else if ((strcasecmp(token, "sun") == 0) ||
1797             (strcasecmp(token, "cc") == 0))
1798         compiler->c_style = SUN;
1799     else
1800         errx(1, "unknown compiler style: %s", token);
1801
1802     if (tspec != NULL)
1803         errx(1, "Excess tokens in compiler: %s", spec);
1804
1805     return (0);
1806 }
1807
1808 #endif /* ! codereview */
1809 int
1810 main(int argc, char **argv)
1811 {
1812     int ch;
1813     cw_compiler_t primary = { NULL, NULL, 0 };
1814     cw_compiler_t shadows[10];
1815     int nshadows = 0;
1816     cw_ictx_t *ctx = newictx();
1817     cw_ictx_t *ctx_shadow = newictx();

```

```

1623     const char *dir;
1624     int do_serial, do_shadow;
1625     int ret = 0;
1626     boolean_t do_serial = B_FALSE;
1627     boolean_t do_exec = B_FALSE;
1628     boolean_t vflg = B_FALSE;
1629     boolean_t cflg = B_FALSE;
1630     boolean_t nflg = B_FALSE;
1631
1632     cw_ictx_t *main_ctx;
1633
1634     static struct option longopts[] = {
1635         { "compiler", no_argument, NULL, 'c' },
1636         { "noecho", no_argument, NULL, 'n' },
1637         { "primary", required_argument, NULL, 'p' },
1638         { "shadow", required_argument, NULL, 's' },
1639         { "versions", no_argument, NULL, 'v' },
1640         { NULL, 0, NULL, 0 }
1641     };
1642 #endif /* ! codereview */
1643
1644     if ((progname = strrchr(argv[0], '/')) == NULL)
1645         progname = argv[0];
1646     else
1647         progname++;
1648
1649     if ((main_ctx = newictx()) == NULL)
1650         if (ctx == NULL || ctx_shadow == NULL)
1651             nomem();
1652
1653     ctx->i_flags = CW_F_ECHO|CW_F_XLATE;
1654
1655     while ((ch = getopt_long(argc, argv, "C", longopts, NULL)) != -1) {
1656         switch (ch) {
1657             case 'c':
1658                 cflg = B_TRUE;
1659                 break;
1660             case 'C':
1661                 Cflg = B_TRUE;
1662                 break;
1663             case 'n':
1664                 nflg = B_TRUE;
1665                 break;
1666             case 'p':
1667                 if (primary.c_path != NULL) {
1668                     warnx("Only one primary compiler may "
1669                          "be specified");
1670                     usage();
1671                 }
1672                 /*
1673                  * Figure out where to get our tools from. This depends on
1674                  * the environment variables set at run time.
1675                  */
1676                 if ((dir = getenv("SPRO_VROOT")) != NULL) {
1677                     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1678                                     "%s/bin", dir);
1679                 } else if ((dir = getenv("SPRO_ROOT")) != NULL) {
1680                     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1681                                     "%s/SS12/bin", dir);
1682                 } else if ((dir = getenv("BUILD_TOOLS")) != NULL) {
1683                     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1684                                     "%s/SUNWspro/SS12/bin", dir);
1685                 }
1686
1687                 if (parse_compiler(optarg, &primary) != 0)
1688                     errx(1, "Couldn't parse %s as a compiler spec",

```

```

1856         optarg);
1857     break;
1858     case 's':
1859         if (nshadows >= 10)
1860             errx(1, "May only use 10 shadows at "
1861                 "the moment");
1862         if (parse_compiler(optarg, &shadows[nshadows]) != 0)
1863             errx(1, "Couldn't parse %s as a compiler spec",
1864                 optarg);
1865         nshadows++;
1866         break;
1867     case 'v':
1868         vflg = B_TRUE;
1869         break;
1870     default:
1871         (void) fprintf(stderr, "Did you forget '--'? \n");
1872         usage();
1873     }
1874     if ((dir = getenv("GCC_ROOT")) != NULL) {
1875         (void) snprintf(default_dir[CW_C_GCC], MAXPATHLEN,
1876             "%s/bin", dir);
1877     }
1878     if (primary.c_path == NULL) {
1879         warnx("A primary compiler must be specified");
1880         usage();
1881     }
1882     do_shadow = (getenv("CW_NO_SHADOW") ? 0 : 1);
1883     do_serial = (getenv("CW_SHADOW_SERIAL") ? 1 : 0);
1884     do_serial = (getenv("CW_SHADOW_SERIAL") == NULL) ? B_FALSE : B_TRUE;
1885     do_exec = (getenv("CW_NO_EXEC") == NULL) ? B_TRUE : B_FALSE;
1886     if (getenv("CW_NO_EXEC") == NULL)
1887         ctx->i_flags |= CW_F_EXEC;
1888     /* Leave room for argv[0] */
1889     argc -= (optind - 1);
1890     argv += (optind - 1);
1891     main_ctx->i_oldargc = argc;
1892     main_ctx->i_oldargv = argv;
1893     main_ctx->i_flags = CW_F_XLATE;
1894     if (nflg == 0)
1895         main_ctx->i_flags |= CW_F_ECHO;
1896     if (do_exec)
1897         main_ctx->i_flags |= CW_F_EXEC;
1898     if (Cflg)
1899         main_ctx->i_flags |= CW_F_CXX;
1900     main_ctx->i_compiler = &primary;
1901     /*
1902      * The first argument must be one of "-_cc", "-_gcc", "-_CC", or "-_g++"
1903      */
1904     if (argc == 1)
1905         usage();
1906     argc--;
1907     argv++;
1908     if (strcmp(argv[0], "-_cc") == 0) {
1909         ctx->i_compiler = CW_C_CC;
1910     } else if (strcmp(argv[0], "-_gcc") == 0) {
1911         ctx->i_compiler = CW_C_GCC;
1912     } else if (strcmp(argv[0], "-_CC") == 0) {
1913         ctx->i_compiler = CW_C_CC;
1914         ctx->i_flags |= CW_F_CXX;
1915     } else if (strcmp(argv[0], "-_g++") == 0) {
1916         ctx->i_compiler = CW_C_GCC;
1917         ctx->i_flags |= CW_F_CXX;
1918     }

```

```

1679     } else {
1680         /* assume "-_gcc" by default */
1681         argc++;
1682         argv--;
1683         ctx->i_compiler = CW_C_GCC;
1684     }
1685     if (cflg) {
1686         (void) fputs(primary.c_path, stdout);
1687         /*
1688          * -_compiler - tell us the path to the primary compiler only
1689          */
1690         if (argc > 1 && strcmp(argv[1], "-_compiler") == 0) {
1691             ctx->i_flags &= ~CW_F_XLATE;
1692             prepcx(ctx);
1693             (void) printf("%s\n", ctx->i_ae->ael_head->ae_arg);
1694             return (0);
1695         }
1696     }
1697     if (vflg) {
1698         (void) printf("cw version %s\n", CW_VERSION);
1699         /*
1700          * -_versions - tell us the cw version, paths to all compilers, and
1701          * ask each for its version if we know how.
1702          */
1703         if (argc > 1 && strcmp(argv[1], "-_versions") == 0) {
1704             (void) printf("cw version %s", CW_VERSION);
1705             if (!do_shadow)
1706                 (void) printf(" (SHADOW MODE DISABLED)");
1707             (void) printf("\n");
1708             (void) fflush(stdout);
1709             main_ctx->i_flags &= ~CW_F_ECHO;
1710             main_ctx->i_flags |= CW_F_PROG|CW_F_EXEC;
1711             ctx->i_flags &= ~CW_F_ECHO;
1712             ctx->i_flags |= CW_F_PROG|CW_F_EXEC;
1713             argc--;
1714             argv++;
1715             do_serial = 1;
1716         }
1717     }
1718     ret |= exec_ctx(main_ctx, do_serial);
1719     for (int i = 0; i < nshadows; i++) {
1720         cw_ictx_t *shadow_ctx;
1721         if ((shadow_ctx = newictx()) == NULL)
1722             nmemem();
1723         memcpy(shadow_ctx, main_ctx, sizeof (cw_ictx_t));
1724         ctx->i_oldargc = argc;
1725         ctx->i_oldargv = argv;
1726         shadow_ctx->i_flags |= CW_F_SHADOW;
1727         shadow_ctx->i_compiler = &shadows[i];
1728         ret |= exec_ctx(ctx, do_serial);
1729     }
1730     /* XXX: Would be nice to run these parallel, too */
1731     ret |= exec_ctx(shadow_ctx, 1);
1732     if (do_shadow) {
1733         (void) memcpy(ctx_shadow, ctx, sizeof (cw_ictx_t));
1734         ctx_shadow->i_flags |= CW_F_SHADOW;
1735         ret |= exec_ctx(ctx_shadow, 1);
1736     }
1737     if (!do_serial)
1738         ret |= reap(main_ctx);

```

```
1725         ret /= reap(ctx);  
1931         return (ret);  
1932 }  
_____unchanged_portion_omitted_____
```

new/usr/src/tools/env/illumos.sh

1

```
*****
9571 Thu Feb 22 01:20:21 2018
new/usr/src/tools/env/illumos.sh
9128 cw(lonbld) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonbld) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
_____unchanged_portion_omitted_____

89 maxjobs DMAKE_MAX_JOBS # "DMAKE_MAX_JOBS" passed as ksh(1) name reference
90 export DMAKE_MAX_JOBS

92 # path to onbld tool binaries
93 ONBLD_BIN='/opt/onbld/bin'

95 # PARENT_WS is used to determine the parent of this workspace. This is
96 # for the options that deal with the parent workspace (such as where the
97 # proto area will go).
98 export PARENT_WS=''

100 # CLONE_WS is the workspace nightly should do a bringover from.
101 export CLONE_WS='ssh://anonhg@hg.illumos.org/illumos-gate'

103 # The bringover, if any, is done as STAFFER.
104 # Set STAFFER to your own login as gatekeeper or developer
105 # The point is to use group "staff" and avoid referencing the parent
106 # workspace as root.
107 # Some scripts optionally send mail messages to MAILTO.
108 #
109 export STAFFER="$LOGNAME"
110 export MAILTO="$STAFFER"

112 # If you wish the mail messages to be From: an arbitrary address, export
113 # MAILFROM.
114 #export MAILFROM="user@example.com"

116 # The project (see project(4)) under which to run this build. If not
117 # specified, the build is simply run in a new task in the current project.
118 export BUILD_PROJECT=''

120 # You should not need to change the next three lines
121 export ATLOG="$CODEMGR_WS/log"
122 export LOGFILE="$ATLOG/nightly.log"
123 export MACH="$(uname -p)"

125 #
126 # The following macro points to the closed binaries. Once illumos has
127 # totally freed itself, we can remove this reference.
128 #
129 # Location of encumbered binaries.
130 export ON_CLOSED_BINS="$CODEMGR_WS/closed"

132 # REF_PROTO_LIST - for comparing the list of stuff in your proto area
133 # with. Generally this should be left alone, since you want to see differences
134 # from your parent (the gate).
135 #
136 export REF_PROTO_LIST="$PARENT_WS/usr/src/proto_list_${MACH}"

139 export ROOT="$CODEMGR_WS/proto/root_${MACH}"
140 export SRC="$CODEMGR_WS/usr/src"
```

new/usr/src/tools/env/illumos.sh

2

```
141 export MULTI_PROTO="no"

143 #
144 # build environment variables, including version info for mcs, motd,
145 # motd, uname and boot messages. Mostly you shouldn't change this except
146 # when the release slips (nah) or you move an environment file to a new
147 # release
148 #
149 export VERSION="'git describe --long --all HEAD | cut -d/ -f2-'"

151 #
152 # the RELEASE and RELEASE_DATE variables are set in Makefile.master;
153 # there might be special reasons to override them here, but that
154 # should not be the case in general
155 #
156 # export RELEASE='5.11'
157 # export RELEASE_DATE='October 2007'

159 # proto area in parent for optionally depositing a copy of headers and
160 # libraries corresponding to the protolibs target
161 # not applicable given the NIGHTLY_OPTIONS
162 #
163 export PARENT_ROOT="$PARENT_WS/proto/root_${MACH}"
164 export PARENT_TOOLS_ROOT="$PARENT_WS/usr/src/tools/proto/root_${MACH}-nd"

166 # Package creation variables. You probably shouldn't change these,
167 # either.
168 #
169 # PKGARCHIVE determines where the repository will be created.
170 #
171 # PKGPUBLISHER_REDIST controls the publisher setting for the repository.
172 #
173 export PKGARCHIVE="${CODEMGR_WS}/packages/${MACH}/nightly"
174 # export PKGPUBLISHER_REDIST='on-redist'

176 # Package manifest format version.
177 export PKGFMT_OUTPUT='v1'

179 # we want make to do as much as it can, just in case there's more than
180 # one problem.
181 export MAKEFLAGS='k'

183 # Magic variables to prevent the devpro compilers/teamware from checking
184 # for updates or sending mail back to devpro on every use.
185 export SUNW_NO_UPDATE_NOTIFY='1'
186 export UT_NO_USAGE_TRACKING='1'

188 # Build tools - don't change these unless you know what you're doing. These
189 # variables allows you to get the compilers and onbld files locally.
190 # Set BUILD_TOOLS to pull everything from one location.
191 # Alternately, you can set ONBLD_TOOLS to where you keep the contents of
192 # SUNWonbld and SPRO_ROOT to where you keep the compilers. SPRO_ROOT
193 # exists to make it easier to test new versions of the compiler.
194 export BUILD_TOOLS='/opt'
195 #export ONBLD_TOOLS='/opt/onbld'
196 export SPRO_ROOT='/opt/SUNWspro'
197 export SPRO_VROOT="$SPRO_ROOT"

199 # Compilers may be specified using the following variables:
200 # PRIMARY_CC - primary C compiler
201 # PRIMARY_CCC - primary C++ compiler
202 #
203 # SHADOW_CCs - list of shadow C compilers
204 # SHADOW_CCCs - list of shadow C++ compilers
205 #
206 # Each entry has the form <name>,<path to binary>,<style> where name is a
```

```
207 # free-form name (possibly used in the makefiles to guard options), path is
208 # the path to the executable. style is the 'style' of command line taken by
209 # the compiler, currently either gnu (or gcc) or sun (or cc), which is also
210 # used by Makefiles to guard options.
211 #
212 # __SUNC and __GNUCC must still be set to reflect the style of the primary
213 # compiler (and to influence the default primary, otherwise)
214 #
215 # for example:
216 # export PRIMARY_CC=gcc4,/opt/gcc/4.4.4/bin/gcc,gnu
217 # export PRIMARY_CCC=gcc4,/opt/gcc/4.4.4/bin/g++,gnu
218 # export SHADOW_CCS=studio12,/opt/SUNWspro/bin/cc,sun
219 # export SHADOW_CCCS=studio12,/opt/SUNWspro/bin/CC,sun
220 #
221 # There can be several space-separated entries in SHADOW_* to run multiple
222 # shadow compilers.
223 #
224 # To disable shadow compilation, unset SHADOW_* or set them to the empty string.
225 #
199 # Disable shadow compilation by default.
200 export CW_NO_SHADOW='1'

227 # This goes along with lint - it is a series of the form "A [y|n]" which
228 # means "go to directory A and run 'make lint'" Then mail me (y) the
229 # difference in the lint output. 'y' should only be used if the area you're
230 # linting is actually lint clean or you'll get lots of mail.
231 # You shouldn't need to change this though.
232 #export LINTDIRS="$SRC y"

234 # Set this flag to 'n' to disable the use of 'checkpaths'. The default,
235 # if the 'N' option is not specified, is to run this test.
236 #CHECK_PATHS='y'

238 # POST_NIGHTLY can be any command to be run at the end of nightly. See
239 # nightly(1) for interactions between environment variables and this command.
240 #POST_NIGHTLY=

242 # Comment this out to disable support for SMB printing, i.e. if you
243 # don't want to bother providing the CUPS headers this needs.
244 export ENABLE_SMB_PRINTING=

246 # If your distro uses certain versions of Perl, make sure either Makefile.master
247 # contains your new defaults OR your .env file sets them.
248 # These are how you would override for building on Omnios r151012, for example.
249 #export PERL_VERSION=5.16.1
250 #export PERL_ARCH=i86pc-solaris-thread-multi-64int
251 #export PERL_PKGVERS=-5161
```

new/usr/src/tools/ndrngen/ndrngen.sh

1

```
*****
2842 Thu Feb 22 01:20:22 2018
new/usr/src/tools/ndrngen/ndrngen.sh
9128 cw(lonblid) should be able to run multiple shadows
9129 file-locking tests shouldn't build multiple source files in one compiler in
9130 DTrace tst.gcc.d isn't useful
9132 cw(lonblid) shouldn't shadow pure preprocessing
Reviewed by: Toomas Soome <tsoome@me.com>
Reviewed? by: Yuri Pankov <yuripv@yuripv.net>
Reviewed? by: Robert Mustacchi <rm@joyent.com>
Reviewed? by: Jason King <jason.king@joyent.com>
*****
_____unchanged_portion_omitted_____

57 if [[ $# -lt 1 ]] ; then
58     ndrngen_usage
59 fi

61 while getopts "Y" FLAG $* ; do
62     case $FLAG in
63         Y)
64             CC_FLAG="y"
65             ;;
66         *)
67             ndrngen_usage
68             ;;
69         esac
70 done

72 if [[ $CC_FLAG = "y" ]] ; then
73     shift $(( $OPTIND - 1 ))

75     if [[ $# -lt 1 ]] ; then
76         ndrngen_usage "C pre-processor path is missing"
77     else
78         CC=$1
79         shift $(( $OPTIND - 1 ))

81         # Check for cw being invoked with -cc or -gcc
82         if [[ $1 = "-_cc" || $1 = "-_gcc" ]] ; then
83             CC_ARG=$1
84             shift $(( $OPTIND - 1 ))
85         fi
86     fi

88 if [[ $CC = "" ]] ; then
89     ndrngen_usage "C pre-processor is not defined"
90 fi

92 # Remove the non-cw options
93 CC=${CC% -- *}

94 if [ ! -f ${CC% *} ] || [ ! -x ${CC% *} ] ; then
95 if [ ! -f $CC ] || [ ! -x $CC ] ; then
96     ndrngen_usage "cannot run $CC"
97 fi

98 for i
99 do
100     if [ ! -r $i ] ; then
101         print "$PROGNAME: ERROR: cannot read $i"
102         exit 1
103     fi
104 fi

105 BASENAME=`basename $i .ndl`
```

new/usr/src/tools/ndrngen/ndrngen.sh

2

```
102     TMP_NAME=$BASENAME.ndl.c
103
104     cp $i $TMP_NAME

106     if $CC -E -D __a64 -D __EXTENSIONS__ -D FILE_OFFSET_BITS=64 \
107     if $CC $CC_ARG -E -D __a64 -D __EXTENSIONS__ -D FILE_OFFSET_BITS=64 \
108     -I. -I${INCDIR} -I${INCDIR}/ndl -DNRGEN $TMP_NAME | \
109     $NDRPROG > $BASENAME.raw
110     then
111         touch ${BASENAME}_ndr.c
112         ndrngen_copy_header $i ${BASENAME}_ndr.c

113         cat - << EOF >> ${BASENAME}_ndr.c
114     /*
115     * Please do not edit this file.
116     * It was generated using ndrngen.
117     */

119 #include <strings.h>
120 #include <smbstrv/ndr.h>
121 #include <smbstrv/ndl/${BASENAME}.ndl>
122 EOF

124     cat $BASENAME.raw >> ${BASENAME}_ndr.c

126     rm -f $BASENAME.raw
127     rm -f $TMP_NAME
128     else
129         rm -f $BASENAME.raw
130         rm -f $TMP_NAME
131         exit 1
132     fi
133 done
```