

new/usr/src/Makefile

1

```
*****
7465 Wed Oct 3 20:38:50 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 2018 Joyent, Inc.
29 # Copyright 2017 Joyent, Inc.
30 # Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
31 #
32 #
33 # Makefile for system source
34 #
35 # include global definitions
36 include Makefile.master
37 #
38 # the Targetdirs file is the AT&T target.dirs file in a makefile format.
39 # it defines TARGETDIRS and ROOTDIRS.
40 include Targetdirs
41 #
42 COMMON_SUBDIRS= data uts lib cmd ucplib ucblib ucblib psm man test
43 sparc_SUBDIRS= stand
44 i386_SUBDIRS= grub boot
45 #
46 #
47 # sparc needs to build stand before psm
48 #
49 $(SPARC_BLD)psm: stand
50 #
51 SUBDIRS= $(COMMON_SUBDIRS) $(MACH)_SUBDIRS
52 #
53 HDRSUBDIRS= uts head lib cmd
54 #
55 # UCB headers are bug-for-bug compatible and not checkable against the header
56 # standards.
```

new/usr/src/Makefile

2

```
57 #
58 CHKHDRSUBDIRS= head uts lib
59 #
60 #
61 # Headers that can be built in parallel
62 #
63 PARALLEL_HEADERS = sysheaders userheaders libheaders cmdheaders
64 #
65 #
66 # Directories that can be built in parallel
67 #
68 PARALLEL_DIRS = data uts lib man
69 #
70 # The check target also causes smf(5) service manifests to be validated.
71 CHKMFSUBDIRS= cmd
72 #
73 # And man page formats
74 CHKMANSUBDIRS = man
75 #
76 MSGSUBDIRS= cmd ucblib ucblib data
77 DOMAINS= \
78     SUNW_OST_ADMIN \
79     SUNW_OST_NETRPC \
80     SUNW_OST_OSCMD \
81     SUNW_OST_OSLIB \
82     SUNW_OST_UCBCMD \
83     SUNW_OST_ZONEINFO
84 #
85 MSGDDIRS= $(DOMAINS:%=$(MSGROOT)/%)
86 MSGDIRS= $(MSGROOT) $(MSGDDIRS) $(MSGROOT)/LC_TIME
87 #
88 all := TARGET= all
89 install := TARGET= install
90 install1 := TARGET= install
91 install2 := TARGET= install
92 install_h := TARGET= install_h
93 clean := TARGET= clean
94 clobber := TARGET= clobber
95 check := TARGET= check
96 #
97 .KEEP_STATE:
98 #
99 #
100 # Note: install does not cause a build in pkg. To build packages,
101 # cd pkg and do a 'make install'
102 #
103 #
104 all: mapfiles closedbins sgs .WAIT $(SUBDIRS) pkg
105 #
106 #
107 # The _msg build is a two-step process. First, the _msg dependency
108 # causes recursive makes in $(MSGSUBDIRS), which stages raw message
109 # files in $(ROOT)/catalog. Second, the action from the install
110 # target rule causes those messages to be post-processed from where
111 # they were staged in $(ROOT)/catalog, and the results placed into the
112 # proto area.
113 #
114 # The stage-licenses target causes the license files needed for
115 # packaging to be pulled from $(SRC) and $(CLOSED) and staged in
116 # $(ROOT)/licenses.
117 #
118 install: install1 install2 _msg stage-licenses
119 @cd msg; pwd; $(MAKE) _msg
120 @rm -rf "$(ROOT)/catalog"
121 #
122 stage-licenses: install2
```

new/usr/src/Makefile

3

```

123     @cd pkg; pwd; $(MAKE) stage-licenses
125 install1: mapfiles closedbins sgs
127 install2: install1 .WAIT $(SUBDIRS)
129 _msg: _msgdirs rootdirs FRC
130     @for m in $(MSGSUBDIRS); do \
131         cd $$m; pwd; $(MAKE) _msg; cd ../ \
132     done
134 mapfiles: bldtools
135     @cd common/mapfiles; pwd; $(MAKE) install
137 clean: $(SUBDIRS) head pkg
138 clobber: $(SUBDIRS) head pkg clobber_local
139 clobber_local:
140     @cd tools; pwd; $(MAKE) clobber
141     @cd common/mapfiles; pwd; $(MAKE) clobber
142     @cd msg; pwd; $(MAKE) clobber
144 # If the tarballs are included inside ON_CLOSED_BINS, use those to extract and
145 # preserve the permissions (in case a distro ships them).
147 closedbins: bldtools $(ROOTDIRS) FRC
148     @CLOSED_ROOT="$ON_CLOSED_BINS/root_$(MACH)${RELEASE_BUILD+nd}"; \
149     if [ -f "$$ON_CLOSED_BINS/on-closed-bins${RELEASE_BUILD+nd}.$(MACH).ta
150     $(ECHO) "Extracting tarball $$ON_CLOSED_BINS/on-closed-bins${RE
151     (cd $(CODEMGR_WS); \
152     $(TAR) xjpf $$ON_CLOSED_BINS/on-closed-bins${RELEASE_BUILD+
153     CLOSED_ROOT=$(CODEMGR_WS)/closed/root_$(MACH)${RELEASE_BUILD+
154     fi; \
155     if [ ! -d "$$CLOSED_ROOT" ]; then \
156     $(ECHO) "Error: ON_CLOSED_BINS must point to closed" \
157     "binaries."; \
158     $(ECHO) "root_$(MACH)${RELEASE_BUILD+nd} is not" \
159     "present in $$ON_CLOSED_BINS."; \
160     exit 1; \
161     fi; \
162     $(ECHO) "Copying closed binaries from $$CLOSED_ROOT"; \
163     (cd $$CLOSED_ROOT; \
164     $(TAR) cfx - $(CODEMGR_WS)/exception_lists/closed-bins .) | \
165     (cd $(ROOT); $(TAR) xBpf -); \
166     ( cd $(ROOT); $(CTFSTRIP) $(cd $$CLOSED_ROOT; $(FIND) \
167     ./kernel ./usr/kernel ./platform/*/kernel -type f -a -perm -u+x | \
168     $(EGREP) -vf $(CODEMGR_WS)/exception_lists/closed-bins ) )
170 #
171 # Declare what parts can be built in parallel
172 # DUMMY at the end is used in case macro expansion produces an empty string to
173 # prevent everything going in parallel
174 #
175 .PARALLEL: $(PARALLEL_HEADERS) DUMMY
176 .PARALLEL: $(PARALLEL_DIRS) DUMMY
178 $(SUBDIRS) head pkg: FRC
179     @cd $@; pwd; $(MAKE) $(TARGET)
181 # librpcsvc has a dependency on headers installed by
182 # userheaders, hence the .WAIT before libheaders.
183 sgs: rootdirs .WAIT sysheaders userheaders .WAIT \
184     libheaders cmdheaders
186 #
187 # Top-level setup target to setup the development environment that includes
188 # headers, tools and generated mapfiles. For open-only builds (i.e.: source

```

new/usr/src/Makefile

4

```

189 # trees w/o usr/closed), this also depends on the closedbins target (above)
190 # in order to properly seed the proto area. Note, although the tools are
191 # dependent on a number of constant mapfiles, the tools themselves are
192 # required to build the generated mapfiles.
193 #
194 setup: closedbins bldtools sgs mapfiles
196 bldtools:
197     @cd tools; pwd; $(MAKE) install
199 # /var/mail/:saved is a special case because of the colon in the name.
200 #
201 rootdirs: $(ROOTDIRS)
202     $(INS) -d -m 775 $(ROOT)/var/mail/:saved
204 lint: FRC
205     $(MAKE) -f Makefile.lint
207 _msgdirs:     $(MSGDIRS)
209 $(ROOTDIRS) $(MSGDIRS):
210     $(INS.dir)
212 userheaders: FRC
213     @cd head; pwd; $(MAKE) install_h
215 libheaders: bldtools
216     @cd lib; pwd; $(MAKE) install_h
218 sysheaders: FRC
219     @cd uts; pwd; $(MAKE) install_h
221 cmdheaders: FRC
222     @cd cmd/devfsadm; pwd; $(MAKE) install_h
223     @cd cmd/fm; pwd; $(MAKE) install_h
224     @cd cmd/mdb; pwd; $(MAKE) install_h
226 check: $(CHKHDRSUBDIRS) $(CHKMFSTSUBDIRS) $(CHKMANSUBDIRS)
228 #
229 # Cross-reference customization: skip all of the subdirectories that
230 # don't contain actual source code.
231 #
232 XRPRUNE = pkg prototypes
233 XRINCDIRS = uts/common head ucobhead
235 cscope.out tags: FRC
236     $(XREF) -f -x $@
238 FRC:
240 # used by nightly
240 #
241 # Targets for reporting compiler versions; nightly uses these.
242 #
244 cc-version:
245     @$ (CW) --versions $(CW_CC_COMPILERS) 2>&1
246     @if $(MACH)_CC -_versions >/dev/null 2>/dev/null; then \
247     $(ECHO) 32-bit compiler; \
248     $(ECHO) $(MACH)_CC; \
249     $(MACH)_CC -_versions 2>&1 | \
250     $(EGREP) '^ (cw|cc/gcc/primary|shadow)'; \
251     else \
252     _COMPILER=$(MACH)_CC -_compiler 2>/dev/null || $(TRUE); \
253     if [ -z "$$_COMPILER" ]; then

```

```

253             $(ECHO) No 32-bit compiler found; \
254             exit 1; \
255         else \
256             $(ECHO) 32-bit compiler; \
257             $(ECHO) ${$(MACH)_CC}; \
258             $(ECHO) $$__COMPILER; \
259             ${$(MACH)_CC} -V 2>&1 | head -1; \
260         fi; \
261     fi

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

247 java-version:
248     @if [ -x "$(JAVAC)" ]; then \
249         $(ECHO) $(JAVAC); \
250         $(JAVA_ROOT)/bin/java -fullversion 2>&1 | head -1; \
251     else \
252         $(ECHO) No Java compiler found; \
253         exit 1; \
254     fi

256 openssl-version:
257     @if [ -x "$(OPENSSL)" ]; then \
258         $(ECHO) $(OPENSSL); \
259         $(OPENSSL) version; \
260         $(OPENSSL) version -f | \
261         $(SED) -n '/_API/{s/.*_API/ API;/s/ -.*//;}; \
262     else \
263         $(ECHO) No OpenSSL utility found; \
264     fi

```

new/usr/src/Makefile.master

1

```
*****
35744 Wed Oct 3 20:38:50 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 # Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
31 #
32 #
33 #
34 # Makefile.master, global definitions for system source
35 #
36 ROOT= /proto
37 #
38 #
39 # Adjunct root, containing an additional proto area to be used for headers
40 # and libraries.
41 #
42 ADJUNCT_PROTO=
43 #
44 #
45 # Adjunct for building things that run on the build machine.
46 #
47 NATIVE_ADJUNCT= /usr
48 #
49 #
50 # RELEASE_BUILD should be cleared for final release builds.
51 # NOT_RELEASE_BUILD is exactly what the name implies.
52 #
53 # __GNUC toggles the building of ON components using gcc and related tools.
54 # Normally set to '#', set it to '' to do gcc build.
55 #
56 # The declaration POUND_SIGN is always '#'. This is needed to get around the
57 # make feature that '#' is always a comment delimiter, even when escaped or
```

new/usr/src/Makefile.master

2

```
58 # quoted. We use this macro expansion method to get POUND_SIGN rather than
59 # always breaking out a shell because the general case can cause a noticeable
60 # slowdown in build times when so many Makefiles include Makefile.master.
61 #
62 # While the majority of users are expected to override the setting below
63 # with an env file (via nightly or bldenv), if you aren't building that way
64 # (ie, you're using "ws" or some other bootstrapping method) then you need
65 # this definition in order to avoid the subshell invocation mentioned above.
66 #
67 #
68 PRE_POUND= pre\#
69 POUND_SIGN= $(PRE_POUND:pre\#=%)
70 #
71 NOT_RELEASE_BUILD=
72 RELEASE_BUILD= $(POUND_SIGN)
73 $(RELEASE_BUILD)NOT_RELEASE_BUILD= $(POUND_SIGN)
74 PATCH_BUILD= $(POUND_SIGN)
75 #
76 # SPARC_BLD is '#' for an Intel build.
77 # INTEL_BLD is '#' for a Sparc build.
78 SPARC_BLD_1= $(MACH:i386=$(POUND_SIGN))
79 SPARC_BLD= $(SPARC_BLD_1:sparc=)
80 INTEL_BLD_1= $(MACH:sparc=$(POUND_SIGN))
81 INTEL_BLD= $(INTEL_BLD_1:i386=)
82 #
83 # The variables below control the compilers used during the build.
84 # There are a number of permutations.
85 #
86 # __GNUC and __SUNC control (and indicate) the primary compiler. Whichever
87 # one is not POUND_SIGN is the primary, with the other as the shadow. They
88 # may also be used to control entirely compiler-specific Makefile assignments.
89 # __GNUC and GCC are the default.
90 #
91 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
92 # There is no Sun C analogue.
93 #
94 # The following version-specific options are operative regardless of which
95 # compiler is primary, and control the versions of the given compilers to be
96 # used. They also allow compiler-version specific Makefile fragments.
97 #
98 #
99 __SUNC= $(POUND_SIGN)
100 $(__SUNC)__GNUC= $(POUND_SIGN)
101 __GNUC64= $(__GNUC)
102 #
103 # Allow build-time "configuration" to enable or disable some things.
104 # The default is POUND_SIGN, meaning "not enabled". If the environment
105 # passes in an override like ENABLE_SMB_PRINTING= (empty) that will
106 # uncomment things in the lower Makefiles to enable the feature.
107 ENABLE_SMB_PRINTING= $(POUND_SIGN)
108 #
109 # CLOSED is the root of the tree that contains source which isn't released
110 # as open source
111 CLOSED= $(SRC)/../closed
112 #
113 # BUILD_TOOLS is the root of all tools including compilers.
114 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.
115 #
116 BUILD_TOOLS= /ws/onnv-tools
117 ONBLD_TOOLS= $(BUILD_TOOLS)/onbld
118 #
119 # define runtime JAVA_HOME, primarily for cmd/pools/poold
120 JAVA_HOME= /usr/java
121 # define buildtime JAVA_ROOT
122 JAVA_ROOT= /usr/java
123 # Build uses java7 by default. Pass one the variables below set to empty
```

new/usr/src/Makefile.master

```

124 # string in the environment to override.
125 BLD_JAVA_6= $(POUND_SIGN)
126 BLD_JAVA_8= $(POUND_SIGN)

128 GNUC_ROOT= /opt/gcc/4.4.4
129 GCCLIBDIR= $(GNUC_ROOT)/lib
130 GCCLIBDIR64= $(GNUC_ROOT)/lib/$(MACH64)

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

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

```

3

new/usr/src/Makefile.master

```

190 PERL_VERSION= 5.10.0
191 PERL_PKGVERS= -510
192 PERL_ARCH = i86pc-solaris-64int
193 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
194 PYTHON_VERSION= 2.7
195 PYTHON_PKGVERS= -27
196 PYTHON= /usr/bin/python$(PYTHON_VERSION)
197 SORT= /usr/bin/sort
198 TOUCH= /usr/bin/touch
199 WC= /usr/bin/wc
200 XARGS= /usr/bin/xargs
201 ELFEDIT= /usr/bin/elfedit
202 DTRACE= /usr/sbin/dtrace -xnolib
203 UNIQ= /usr/bin/uniq
204 TAR= /usr/bin/tar
205 ASTBINDIR= /usr/ast/bin
206 MSGCC= $(ASTBINDIR)/msgcc
207 MSGFMT= /usr/bin/msgfmt -s
208 LCDEF= $(ONBLD_TOOLS)/bin/$(MACH)/localedef
209 TIC= $(ONBLD_TOOLS)/bin/$(MACH)/tic
210 ZIC= $(ONBLD_TOOLS)/bin/$(MACH)/zic
211 OPENSSEL= /usr/bin/openssl

213 FILEMODE= 644
214 DIRMODE= 755

216 # Declare that nothing should be built in parallel.
217 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
218 .NO_PARALLEL:

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

231 DOT_H_CHECK= \
232 @$ (ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL); \
233 $(HDRCHK) $< $(HDRCHK_TAIL)

235 DOT_X_CHECK= \
236 @$ (ECHO) "checking $<"; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
237 $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

239 DOT_C_CHECK= \
240 @$ (ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL)

242 MANIFEST_CHECK= \
243 @$ (ECHO) "checking $<"; \
244 SVCCFG_DTD=$(SRC)/cmd/svc/dtd/service_bundle.dtd.1 \
245 SVCCFG_REPOSITORY=$(SRC)/cmd/svc/seed/global.db \
246 SVCCFG_CONFIGD_PATH=$(SRC)/cmd/svc/configd/svc.configd-native \
247 $(SRC)/cmd/svc/svccfg/svccfg-native validate $<

249 INS.file= $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
250 INS.dir= $(INS) -s -d -m $(DIRMODE) $@
251 # installs and renames at once
252 #
253 INS.rename= $(INS.file); $(MV) $(@D)/$(<F) $@

255 # install a link

```

4

```

256 INSLINKTARGET= $<
257 INS.link= $(RM) $@; $(LN) $(INSLINKTARGET) $@
258 INS.symlink= $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

260 #
261 # Python bakes the mtime of the .py file into the compiled .pyc and
262 # rebuilds if the baked-in mtime != the mtime of the source file
263 # (rather than only if it's less than), thus when installing python
264 # files we must make certain to not adjust the mtime of the source
265 # (.py) file.
266 #
267 INS.pyfile= $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $

269 # MACH must be set in the shell environment per uname -p on the build host
270 # More specific architecture variables should be set in lower makefiles.
271 #
272 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
273 # architectures on which we do not build 64-bit versions.
274 # (There are no such architectures at the moment.)
275 #
276 # Set BUILD64=# in the environment to disable 64-bit amd64
277 # builds on i386 machines.

279 MACH64_1= $(MACH:sparc=sparcv9)
280 MACH64= $(MACH64_1:i386=amd64)

282 MACH32_1= $(MACH:sparc=sparcv7)
283 MACH32= $(MACH32_1:i386=i86)

285 sparc_BUILD64=
286 i386_BUILD64=
287 BUILD64= $($(_MACH)_BUILD64)

289 #
290 # C compiler mode. Future compilers may change the default on us,
291 # so force extended ANSI mode globally. Lower level makefiles can
292 # override this by setting CCMODE.
293 #
294 CCMODE= -Xa
295 CCMODE64= -Xa

297 #
298 # C compiler verbose mode. This is so we can enable it globally,
299 # but turn it off in the lower level makefiles of things we cannot
300 # (or aren't going to) fix.
301 #
302 CCVERBOSE= -v

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

308 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
309 # symbols (used to detect conflicts between objects that use global registers)
310 # we disable this now for safety, and because genunix doesn't link with
311 # this feature (the v9 default) enabled.
312 #
313 # REGSYM is separate since the C++ driver syntax is different.
314 CCREGSYM= -Wc,-Qiselect-regsym=0
315 CCCREGSYM= -Qoption cg -Qiselect-regsym=0

317 # Prevent the removal of static symbols by the SPARC code generator (cg).
318 # The x86 code generator (ube) does not remove such symbols and as such
319 # using this workaround is not applicable for x86.
320 #
321 CCSTATICSYM= -Wc,-Qassembler-ounrefsym=0

```

```

322 #
323 # generate 32-bit addresses in the v9 kernel. Saves memory.
324 CCABS32= -Wc,-xcode=abs32
325 #
326 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
327 # system calls.
328 CC32BITCALLERS= -_gcc=-massume-32bit-callers

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

339 # One optimization the compiler might perform is to turn this:
340 # #pragma weak foo
341 # extern int foo;
342 # if (&foo)
343 #     foo = 5;
344 # into
345 #     foo = 5;
346 # Since we do some of this (foo might be referenced in common kernel code
347 # but provided only for some cpu modules or platforms), we disable this
348 # optimization.
349 #
350 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
351 i386_CCUNBOUND =
352 CCUNBOUND = $($(_MACH)_CCUNBOUND)

354 #
355 # compiler '-xarch' flag. This is here to centralize it and make it
356 # overridable for testing.
357 sparc_XARCH= -m32
358 sparcv9_XARCH= -m64
359 i386_XARCH= -m32
360 amd64_XARCH= -m64 -Ui386 -U__i386

362 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
363 sparc_AS_XARCH= -xarch=v8plus
364 sparcv9_AS_XARCH= -xarch=v9
365 i386_AS_XARCH=
366 amd64_AS_XARCH= -xarch=amd64 -P -Ui386 -U__i386

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

383 SAVEARGS= -Wu,-save_args
384 amd64_STAND_FLAGS += $(SAVEARGS)

386 STAND_FLAGS_32 = $($(_MACH)_STAND_FLAGS)
387 STAND_FLAGS_64 = $($(_MACH64)_STAND_FLAGS)

```

```

389 #
390 # disable the incremental linker
391 ILDOFF= -xildoff
392 #
393 XDEPEND= -xdepend
394 XFFLAG= -xF=%all
395 XESS= -xs
396 XSTRCONST= -xstrconst

398 #
399 # turn warnings into errors (C)
400 CERRWARN = -errtags=yes -errwarn=%all
401 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
402 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

404 CERRWARN += -_gcc=-Wno-missing-braces
405 CERRWARN += -_gcc=-Wno-sign-compare
406 CERRWARN += -_gcc=-Wno-unknown-pragmas
407 CERRWARN += -_gcc=-Wno-unused-parameter
408 CERRWARN += -_gcc=-Wno-missing-field-initializers

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

413 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
414 # -nd builds
415 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
416 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

418 #
419 # turn warnings into errors (C++)
420 CCERRWARN= -xwe

422 # C standard. Keep Studio flags until we get rid of lint.
423 CSTD_GNU89= -xc99=%none
424 CSTD_GNU99= -xc99=%all
425 CSTD= $(CSTD_GNU89)
426 C99LMODE= $(CSTD:-xc99%=-Xc99%)

428 # In most places, assignments to these macros should be appended with +=
429 # (CPPFLAGS.first allows values to be prepended to CPPFLAGS).
430 sparc_CFLAGS= $(sparc_XARCH) $(CCSTATICSYM)
431 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
432 $(CCSTATICSYM)
433 i386_CFLAGS= $(i386_XARCH)
434 amd64_CFLAGS= $(amd64_XARCH)

436 sparc_ASFLAGS= $(sparc_AS_XARCH)
437 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
438 i386_ASFLAGS= $(i386_AS_XARCH)
439 amd64_ASFLAGS= $(amd64_AS_XARCH)

441 #
442 sparc_COPTFLAG= -xO3
443 sparcv9_COPTFLAG= -xO3
444 i386_COPTFLAG= -O
445 amd64_COPTFLAG= -xO3

447 COPTFLAG= $($(_MACH)_COPTFLAG)
448 COPTFLAG64= $($(_MACH64)_COPTFLAG)

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

```

```

454 # Direct the Sun Studio compiler to use a static globalization prefix based on t
455 # name of the module rather than something unique. Otherwise, objects
456 # will not build deterministically, as subsequent compilations of identical
457 # source will yield objects that always look different.
458 #
459 # In the same spirit, this will also remove the date from the N_OPT stab.
460 CGLOBALSTATIC= -W0,-xglobalstatic

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

466 #
467 # Default debug format for Sun Studio 11 is dwarf, so force it to
468 # generate stabs.
469 #
470 DEBUGFORMAT= -xdebugformat=stabs

472 #
473 # Flags used to build in debug mode for ctf generation. Bugs in the Devpro
474 # compilers currently prevent us from building with cc-emitted DWARF.
475 #
476 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)
477 CTF_FLAGS_i386 = -g $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)

479 CTF_FLAGS_sparcv9 = $(CTF_FLAGS_sparc)
480 CTF_FLAGS_amd64 = $(CTF_FLAGS_i386)

482 # Sun Studio produces broken userland code when saving arguments.
483 $(__GNUCC)CTF_FLAGS_amd64 += $(SAVEARGS)

485 CTF_FLAGS_32 = $(CTF_FLAGS_${MACH}) $(DEBUGFORMAT)
486 CTF_FLAGS_64 = $(CTF_FLAGS_${MACH64}) $(DEBUGFORMAT)
487 CTF_FLAGS = $(CTF_FLAGS_32)

489 #
490 # Flags used with genoffsets
491 #
492 GOFLAGS = $(CALLSYMS) $(CDWARFSTR)
493 GOFLAGS = -_noecho \
494 $(CALLSYMS) \
495 $(CDWARFSTR)

497 OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
498 $(CW) --noecho $(CW_CC_COMPILERS) -- $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)
499 $(CC) $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)

497 OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
498 $(CW) --noecho $(CW_CC_COMPILERS) -- $(GOFLAGS) $(CFLAGS64) $(CPPFLAGS)
499 $(CC) $(GOFLAGS) $(CFLAGS64) $(CPPFLAGS)

500 #
501 # tradeoff time for space (smaller is better)
502 #
503 sparc_SPACEFLAG = -xspace -W0,-Lt
504 sparcv9_SPACEFLAG = -xspace -W0,-Lt
505 i386_SPACEFLAG = -xspace
506 amd64_SPACEFLAG =

508 SPACEFLAG = $($(_MACH)_SPACEFLAG)
509 SPACEFLAG64 = $($(_MACH64)_SPACEFLAG)

511 #
512 # The Sun Studio 11 compiler has changed the behaviour of integer
513 # wrap arounds and so a flag is needed to use the legacy behaviour
514 # (without this flag panics/hangs could be exposed within the source).

```

```

515 #
516 sparc_IROPTFLAG      = -W2,-xwrap_int
517 sparcv9_IROPTFLAG    = -W2,-xwrap_int
518 i386_IROPTFLAG       =
519 amd64_IROPTFLAG      =

521 IROPTFLAG            = ${$(MACH)_IROPTFLAG}
522 IROPTFLAG64         = ${$(MACH64)_IROPTFLAG}

524 sparc_XREGSFLAG      = -xregs=no%appl
525 sparcv9_XREGSFLAG    = -xregs=no%appl
526 i386_XREGSFLAG       =
527 amd64_XREGSFLAG      =

529 XREGSFLAG           = ${$(MACH)_XREGSFLAG}
530 XREGSFLAG64         = ${$(MACH64)_XREGSFLAG}

532 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
533 # avoids stripping it.
534 SOURCEDEBUG         = $(POUND_SIGN)
535 SRCDBGBLD           = $(SOURCEDEBUG=yes=)

537 #
538 # These variables are intended ONLY for use by developers to safely pass extra
539 # flags to the compilers without unintentionally overriding Makefile-set
540 # flags. They should NEVER be set to any value in a Makefile.
541 #
542 # They come last in the associated FLAGS variable such that they can
543 # explicitly override things if necessary, there are gaps in this, but it's
544 # the best we can manage.
545 #
546 CUSERFLAGS          =
547 CUSERFLAGS64        = $(CUSERFLAGS)
548 CCUSERFLAGS         =
549 CCUSERFLAGS64       = $(CCUSERFLAGS)

551 CSOURCEDEBUGFLAGS   =
552 CCSOURCEDEBUGFLAGS  =
553 $(SRCDBGBLD)CSOURCEDEBUGFLAGS = -g -xs
554 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS = -g -xs

556 CFLAGS=             $(COPTFLAG) ${$(MACH)_CFLAGS} $(SPACEFLAG) $(CCMODE) \
557                     $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG) \
558                     $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
559                     $(CUSERFLAGS)
560 CFLAGS64=           $(COPTFLAG64) ${$(MACH64)_CFLAGS} $(SPACEFLAG64) $(CCMODE64) \
561                     $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG64) \
562                     $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
563                     $(CUSERFLAGS64)
564 #
565 # Flags that are used to build parts of the code that are subsequently
566 # run on the build machine (also known as the NATIVE_BUILD).
567 #
568 NATIVE_CFLAGS=      $(COPTFLAG) ${$(NATIVE_MACH)_CFLAGS} $(CCMODE) \
569                     $(ILDOFF) $(CERRWARN) $(CSTD) ${$(NATIVE_MACH)_CCUNBOUND} \
570                     $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
571                     $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

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

```

```

581 CPPFLAGS=           $(CPPFLAGS.first) $(CPPFLAGS.master)
582 AS_CPPFLAGS=        $(CPPFLAGS.first) $(CPPFLAGS.master)
583 JAVAFLAGS=          -source 1.6 -target 1.6 -Xlint:deprecation,-options

585 #
586 # For source message catalogue
587 #
588 .SUFFIXES: $(SUFFIXES) .i .po
589 MSGROOT= $(ROOT)/catalog
590 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
591 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
592 DCMMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
593 DCMMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

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

607 #
608 # This is overwritten by local Makefile when PROG is a list.
609 #
610 POFILE= $(PROG).po

612 sparc_CCFLAGS=      -cg92 -compat=4 \
613                     -Qoption ccfe -messages=no%anachronism \
614                     $(CCERRWARN)
615 sparcv9_CCFLAGS=    $(sparcv9_XARCH) -dalign -compat=5 \
616                     -Qoption ccfe -messages=no%anachronism \
617                     -Qoption ccfe -features=no%conststrings \
618                     $(CCREGSYM) \
619                     $(CCERRWARN)
620 i386_CCFLAGS=       -compat=4 \
621                     -Qoption ccfe -messages=no%anachronism \
622                     -Qoption ccfe -features=no%conststrings \
623                     $(CCERRWARN)
624 amd64_CCFLAGS=      $(amd64_XARCH) -compat=5 \
625                     -Qoption ccfe -messages=no%anachronism \
626                     -Qoption ccfe -features=no%conststrings \
627                     $(CCERRWARN)

629 sparc_CCOPTFLAG=    -O
630 sparcv9_CCOPTFLAG=  -O
631 i386_CCOPTFLAG=     -O
632 amd64_CCOPTFLAG=    -O

634 CCOPTFLAG=         ${$(MACH)_CCOPTFLAG}
635 CCOPTFLAG64=       ${$(MACH64)_CCOPTFLAG}
636 CCFLAGS=           $(CCOPTFLAG) ${$(MACH)_CCFLAGS} $(CCSOURCEDEBUGFLAGS) \
637                     $(CUSERFLAGS)
638 CCFLAGS64=         $(CCOPTFLAG64) ${$(MACH64)_CCFLAGS} $(CCSOURCEDEBUGFLAGS) \
639                     $(CCUSERFLAGS64)

641 #
642 #
643 #
644 ELFWRAP_FLAGS =
645 ELFWRAP_FLAGS64 = -64

```



```

647 #
648 # Various mapfiles that are used throughout the build, and delivered to
649 # /usr/lib/ld.
650 #
651 MAPFILE.NED_i386 = $(SRC)/common/mapfiles/common/map.noexdata
652 MAPFILE.NED_sparc =
653 MAPFILE.NED = $(MAPFILE.NED_$(MACH))
654 MAPFILE.PGA = $(SRC)/common/mapfiles/common/map.pagealign
655 MAPFILE.NES = $(SRC)/common/mapfiles/common/map.noexstk
656 MAPFILE.FLT = $(SRC)/common/mapfiles/common/map.filter
657 MAPFILE.LEX = $(SRC)/common/mapfiles/common/map.lex.yy

659 #
660 # Generated mapfiles that are compiler specific, and used throughout the
661 # build. These mapfiles are not delivered in /usr/lib/ld.
662 #
663 MAPFILE.NGB_sparc= $(SRC)/common/mapfiles/gen/sparc_cc_map.noexglobs
664 $(__GNUC64)MAPFILE.NGB_sparc= \
665 $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexglobs
666 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexglobs
667 $(__GNUC64)MAPFILE.NGB_sparcv9= \
668 $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexglobs
669 MAPFILE.NGB_i386= $(SRC)/common/mapfiles/gen/i386_cc_map.noexglobs
670 $(__GNUC64)MAPFILE.NGB_i386= \
671 $(SRC)/common/mapfiles/gen/i386_gcc_map.noexglobs
672 MAPFILE.NGB_amd64= $(SRC)/common/mapfiles/gen/amd64_cc_map.noexglobs
673 $(__GNUC64)MAPFILE.NGB_amd64= \
674 $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexglobs
675 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH))

677 #
678 # A generic interface mapfile name, used by various dynamic objects to define
679 # the interfaces and interposers the object must export.
680 #
681 MAPFILE.INT = mapfile-intf

683 #
684 # LDLIBS32 and LDLIBS64 can be set in the environment to override the following
685 # assignments.
686 #
687 # These environment settings make sure that no libraries are searched outside
688 # of the local workspace proto area:
689 # LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
690 # LDLIBS64=-YP,$ROOT/lib:$MACH64:$ROOT/usr/lib:$MACH64
691 #
692 LDLIBS32 = $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
693 LDLIBS32 += $(ADJUNCT_PROTO:%=-L%/usr/lib -L%/lib)
694 LDLIBS.cmd = $(LDLIBS32)
695 LDLIBS.lib = $(LDLIBS32)

697 LDLIBS64 = $(ENVLDLIBS1:%=%/$(MACH64)) \
698 $(ENVLDLIBS2:%=%/$(MACH64)) \
699 $(ENVLDLIBS3:%=%/$(MACH64))
700 LDLIBS64 += $(ADJUNCT_PROTO:%=-L%/usr/lib/$(MACH64) -L%/lib/$(MACH64))

702 #
703 # Define compilation macros.
704 #
705 COMPILER.c= $(CC) $(CFLAGS) $(CPPFLAGS) -c
706 COMPILER64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) $(CPPFLAGS) -c
707 COMPILER.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
708 COMPILER64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(CPPFLAGS) -c
709 COMPILER.s= $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
710 COMPILER64.s= $(AS) $(ASFLAGS) $(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
711 COMPILER.d= $(DTRACE) -G -32
712 COMPILER64.d= $(DTRACE) -G -64

```

```

713 COMPILER.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
714 COMPILER64.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

716 CLASSPATH= .
717 COMPILER.java= $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

719 #
720 # Link time macros
721 #
722 CCNEEDED = -lC
723 CCEXTNEEDED = -lCrun -lCstd
724 $(__GNUC)CCNEEDED = -L$(GCCLIBDIR) -lstc++ -lgcc_s
725 $(__GNUC)CCEXTNEEDED = $(CCNEEDED)

727 LINK.c= $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
728 LINK64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
729 NORUNPATH= -norunpath -nolib
730 LINK.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
731 $(LDFLAGS) $(CCNEEDED)
732 LINK64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
733 $(LDFLAGS) $(CCNEEDED)

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

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

763 SECLEVEL= core
764 LINT.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
765 $(ALWAYS_LINT_DEFS)
766 LINT64.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
767 $(ALWAYS_LINT_DEFS)
768 LINT.s= $(LINT.c)

770 # For some future builds, NATIVE_MACH and MACH might be different.
771 # Therefore, NATIVE_MACH needs to be redefined in the
772 # environment as 'uname -p' to override this macro.
773 #
774 # For now at least, we cross-compile amd64 on i386 machines.
775 NATIVE_MACH= $(MACH:amd64=i386)

777 # Define native compilation macros
778 #

```

```

780 # Base directory where compilers are loaded.
781 # Defined here so it can be overridden by developer.
782 #
783 SPRO_ROOT=          $(BUILD_TOOLS)/SUNwspiro
784 SPRO_VROOT=        $(SPRO_ROOT)/SS12
785 GNU_ROOT=          /usr

787 $(__GNUC)PRIMARY_CC= gcc4,$(GNUC_ROOT)/bin/gcc,gnu
788 $(__SUNC)PRIMARY_CC= studio12,$(SPRO_VROOT)/bin/cc,sun
789 $(__GNUC)PRIMARY_CCC= gcc4,$(GNUC_ROOT)/bin/g++,gnu
790 $(__SUNC)PRIMARY_CCC= studio12,$(SPRO_VROOT)/bin/CC,sun

792 CW_CC_COMPILERS=    $(PRIMARY_CC:%--primary %) $(SHADOW_CCS:%--shadow %)
793 CW_CCC_COMPILERS=  $(PRIMARY_CCC:%--primary %) $(SHADOW_CCCS:%--shadow %)

796 # Till SS12u1 formally becomes the NV CBE, LINT is hard
797 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
798 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
799 # i386_LINT, amd64_LINT.
800 # Reset them when SS12u1 is rolled out.
801 #

803 # Specify platform compiler versions for languages
804 # that we use (currently only c and c++).
805 #
806 CW=                $(ONBLD_TOOLS)/bin/$(MACH)/cw

808 BUILD_CC=          $(CW) $(CW_CC_COMPILERS) --
809 BUILD_CCC=         $(CW) -C $(CW_CCC_COMPILERS) --
810 BUILD_CPP=         /usr/ccs/lib/cpp
811 BUILD_LD=          /usr/ccs/bin/ld
812 BUILD_LINT=        $(SPRO_ROOT)/sunstudio12.1/bin/lint

814 $(MACH)_CC=        $(BUILD_CC)
815 $(MACH)_CCC=       $(BUILD_CCC)
816 $(MACH)_CPP=       $(BUILD_CPP)
817 $(MACH)_LD=        $(BUILD_LD)
818 $(MACH)_LINT=      $(BUILD_LINT)
819 $(MACH64)_CC=      $(BUILD_CC)
820 $(MACH64)_CCC=     $(BUILD_CCC)
821 $(MACH64)_CPP=     $(BUILD_CPP)
822 $(MACH64)_LD=     $(BUILD_LD)
823 $(MACH64)_LINT=    $(BUILD_LINT)

799 sparc_CC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
800 $(__GNUC)sparc_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
801 sparc_CCC=         $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
802 $(__GNUC)sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
803 sparc_CPP=         /usr/ccs/lib/cpp
825 sparc_AS=          /usr/ccs/bin/as -xregsym=no
826 sparcv9_AS=        $(MACH)_AS
805 sparc_LD=          /usr/ccs/bin/ld
806 sparc_LINT=        $(SPRO_ROOT)/sunstudio12.1/bin/lint

808 sparcv9_CC=        $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
809 $(__GNUC64)sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
810 sparcv9_CCC=       $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
811 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
812 sparcv9_CPP=       /usr/ccs/lib/cpp
813 sparcv9_AS=        /usr/ccs/bin/as -xregsym=no
814 sparcv9_LD=        /usr/ccs/bin/ld
815 sparcv9_LINT=      $(SPRO_ROOT)/sunstudio12.1/bin/lint

817 i386_CC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc

```

```

818 $(__GNUC)i386_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
819 i386_CCC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
820 $(__GNUC)i386_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
821 i386_CPP=          /usr/ccs/lib/cpp
828 i386_AS=          /usr/ccs/bin/as
829 $(__GNUC)i386_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
824 i386_LD=          /usr/ccs/bin/ld
825 i386_LINT=         $(SPRO_ROOT)/sunstudio12.1/bin/lint

827 amd64_CC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
828 $(__GNUC64)amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
829 amd64_CCC=        $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
830 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
831 amd64_CPP=         /usr/ccs/lib/cpp
830 amd64_AS=         $(ONBLD_TOOLS)/bin/$(MACH)/aw
833 amd64_LD=         /usr/ccs/bin/ld
834 amd64_LINT=        $(SPRO_ROOT)/sunstudio12.1/bin/lint

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

839 #
840 # Makefile.master.64 overrides these settings
841 #
842 CC=                $(NATIVECC)
843 CCC=               $(NATIVECCC)
844 CPP=               $(NATIVECPP)
845 AS=                $(NATIVEAS)
846 LD=                $(NATIVELD)
847 LINT=              $(NATIVELINT)

853 # The real compilers used for this build
854 CW_CC_CMD=         $(CC) -_compiler
855 CW_CCC_CMD=        $(CCC) -_compiler
856 REAL_CC=           $(CW_CC_CMD:sh)
857 REAL_CCC=          $(CW_CCC_CMD:sh)

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

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

859 ZDEFS=             -zdefs
860 ZDIRECT=            -zdirect
861 ZIGNORE=           -zignore
862 ZINITFIRST=       -zinitfirst
863 ZINTERPOSE=       -zinterpose
864 ZLAZYLOAD=        -zlazyload
865 ZLOADFLTR=        -zloadfltr
866 ZMULDEFS=         -zmuldefs
867 ZNODEFAULTLIB=    -znodefaultlib
868 ZNODEFS=           -znodefs
869 ZNODELETE=         -znodelete
870 ZNODLOPEN=        -znodlopen
871 ZNODUMP=           -znodump
872 ZNOLAZYLOAD=       -znolazyload
873 ZNOLDYNSYM=        -znoldynsym

```

```

874 ZNORELOC=      -znoreloc
875 ZNOVERSION=    -znoversion
876 ZRECORD=       -zrecord
877 ZREDLOCSYM=    -zredlocsym
878 ZTEXT=         -ztext
879 ZVERBOSE=      -zverbose

881 GSHARED=       -G
882 CCMT=         -mt

884 # Handle different PIC models on different ISAs
885 # (May be overridden by lower-level Makefiles)

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

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

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

909 AS_PICFLAGS=           ${C_PICFLAGS}
910 AS_BIGPICFLAGS=        ${C_BIGPICFLAGS}

912 #
913 # Default label for CTF sections
914 #
915 CTFCVTFLAGS=          -i -L VERSION

917 #
918 # Override to pass module-specific flags to ctfmerge.  Currently used only by
919 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
920 # stripping.
921 #
922 CTFMRGFLAGS=

924 CTFCONVERT_O          = ${CTFCONVERT} ${CTFCVTFLAGS} $@

926 # Rules (normally from make.rules) and macros which are used for post
927 # processing files.  Normally, these do stripping of the comment section
928 # automatically.
929 #   RELEASE_CM:      Should be edited to reflect the release.
930 #   POST_PROCESS_O:  Post-processing for '.o' files.
931 #   POST_PROCESS_A:  Post-processing for '.a' files (currently null).
932 #   POST_PROCESS_SO: Post-processing for '.so' files.
933 #   POST_PROCESS:    Post-processing for executable files (no suffix).
934 # Note that these macros are not completely generalized as they are to be
935 # used with the file name to be processed following.
936 #
937 # It is left as an exercise to Release Engineering to embellish the generation
938 # of the release comment string.
939 #

```

```

940 #   If this is a standard development build:
941 #       compress the comment section (mcs -c)
942 #       add the standard comment (mcs -a $(RELEASE_CM))
943 #       add the development specific comment (mcs -a $(DEV_CM))
944 #
945 #   If this is an installation build:
946 #       delete the comment section (mcs -d)
947 #       add the standard comment (mcs -a $(RELEASE_CM))
948 #       add the development specific comment (mcs -a $(DEV_CM))
949 #
950 #   If this is an release build:
951 #       delete the comment section (mcs -d)
952 #       add the standard comment (mcs -a $(RELEASE_CM))
953 #
954 # The following list of macros are used in the definition of RELEASE_CM
955 # which is used to label all binaries in the build:
956 #
957 #   RELEASE           Specific release of the build, eg: 5.2
958 #   RELEASE_MAJOR     Major version number part of $(RELEASE)
959 #   RELEASE_MINOR     Minor version number part of $(RELEASE)
960 #   VERSION           Version of the build (alpha, beta, Generic)
961 #   PATCHID           If this is a patch this value should contain
962 #                     the patchid value (eg: "Generic 100832-01"), otherwise
963 #                     it will be set to $(VERSION)
964 #   RELEASE_DATE      Date of the Release Build
965 #   PATCH_DATE        Date the patch was created, if this is blank it
966 #                     will default to the RELEASE_DATE
967 #
968 RELEASE_MAJOR=       5
969 RELEASE_MINOR=       11
970 RELEASE=             ${RELEASE_MAJOR}.${RELEASE_MINOR}
971 VERSION=             SunOS Development
972 PATCHID=             ${VERSION}
973 RELEASE_DATE=        release date not set
974 PATCH_DATE=         ${RELEASE_DATE}
975 RELEASE_CM=          "@${(POUND_SIGN)}SunOS ${RELEASE} ${PATCHID} ${PATCH_DATE}"
976 DEV_CM=              "@${(POUND_SIGN)}SunOS Internal Development: non-nightly build"

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

981 STRIP_STABS=        ${STRIP} -x $@
982 ${SRCDBGBLD}STRIP_STABS= :

984 POST_PROCESS_O=
985 POST_PROCESS_A=
986 POST_PROCESS_SO=    ${PROCESS_COMMENT} $@ ; ${STRIP_STABS} ; \
987                     ${ELFSIGN_OBJECT}
988 POST_PROCESS=        ${PROCESS_COMMENT} $@ ; ${STRIP_STABS} ; \
989                     ${ELFSIGN_OBJECT}

991 #
992 # chk4ubin is a tool that inspects a module for a symbol table
993 # ELF section size which can trigger an OBP bug on older platforms.
994 # This problem affects only specific sun4u bootable modules.
995 #
996 CHK4UBIN=           ${ONBLD_TOOLS}/bin/${MACH}/chk4ubin
997 CHK4UBINFLAGS=
998 CHK4UBINARY=        ${CHK4UBIN} ${CHK4UBINFLAGS} $@

1000 #
1001 # PKGARCHIVE specifies the default location where packages should be
1002 # placed if built.
1003 #
1004 ${RELEASE_BUILD}PKGARCHIVESUFFIX= -nd
1005 PKGARCHIVE=${SRC}/../..../packages/${MACH}/nightly${PKGARCHIVESUFFIX}

```

```

1007 #
1008 # The repositories will be created with these publisher settings. To
1009 # update an image to the resulting repositories, this must match the
1010 # publisher name provided to "pkg set-publisher."
1011 #
1012 PKGPUBLISHER_REDIST=    on-nightly
1013 PKGPUBLISHER_NONREDIST= on-extra

1015 #      Default build rules which perform comment section post-processing.
1016 #
1017 .c:
1018     $(LINK.c) -o $@ $< $(LDLIBS)
1019     $(POST_PROCESS)
1020 .c.o:
1021     $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1022     $(POST_PROCESS_O)
1023 .c.a:
1024     $(COMPILE.c) -o $% $<
1025     $(PROCESS_COMMENT) $%
1026     $(AR) $(ARFLAGS) $@ $%
1027     $(RM) $%
1028 .s.o:
1029     $(COMPILE.s) -o $@ $<
1030     $(POST_PROCESS_O)
1031 .s.a:
1032     $(COMPILE.s) -o $% $<
1033     $(PROCESS_COMMENT) $%
1034     $(AR) $(ARFLAGS) $@ $%
1035     $(RM) $%
1036 .cc:
1037     $(LINK.cc) -o $@ $< $(LDLIBS)
1038     $(POST_PROCESS)
1039 .cc.o:
1040     $(COMPILE.cc) $(OUTPUT_OPTION) $<
1041     $(POST_PROCESS_O)
1042 .cc.a:
1043     $(COMPILE.cc) -o $% $<
1044     $(AR) $(ARFLAGS) $@ $%
1045     $(PROCESS_COMMENT) $%
1046     $(RM) $%
1047 .y:
1048     $(YACC.y) $<
1049     $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1050     $(POST_PROCESS)
1051     $(RM) y.tab.c
1052 .y.o:
1053     $(YACC.y) $<
1054     $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1055     $(POST_PROCESS_O)
1056     $(RM) y.tab.c
1057 .l:
1058     $(RM) $*.c
1059     $(LEX.l) $< > $*.c
1060     $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1061     $(POST_PROCESS)
1062     $(RM) $*.c
1063 .l.o:
1064     $(RM) $*.c
1065     $(LEX.l) $< > $*.c
1066     $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1067     $(POST_PROCESS_O)
1068     $(RM) $*.c

1070 .bin.o:
1071     $(COMPILE.b) -o $@ $<

```

```

1072     $(POST_PROCESS_O)

1074 .java.class:
1075     $(COMPILE.java) $<

1077 # Bourne and Korn shell script message catalog build rules.
1078 # We extract all gettext strings with sed(1) (being careful to permit
1079 # multiple gettext strings on the same line), weed out the dups, and
1080 # build the catalogue with awk(1).

1082 .sh.po .ksh.po:
1083     $(SED) -n -e ":a"
1084     -e "h"
1085     -e "s/.*gettext *\([^\"]*\).*\1/p"
1086     -e "x"
1087     -e "s/\(.*\)gettext *\"[^\"]*\"(.*)/\1\2/"
1088     -e "t a"
1089     $< | sort -u | $(AWK) '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1091 #
1092 # Python and Perl executable and message catalog build rules.
1093 #
1094 .SUFFIXES: .pl .pm .py .pyc

1096 .pl:
1097     $(RM) $@;
1098     $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1099     $(CHMOD) +x $@

1101 .py:
1102     $(RM) $@; $(SED) -e "1s:^#@PYTHON@:#!$(PYTHON):" < $< > $@; $(CHMOD)

1104 .py.pyc:
1105     $(RM) $@
1106     $(PYTHON) -mpy_compile $<
1107     @[ $(<)c = $@ ] || $(MV) $(<)c $@

1109 .py.po:
1110     $(GNUXGETTEXT) $(GNUXGETTEXTFLAGS) -d $(<F:%.py=) $< ;

1112 .pl.po .pm.po:
1113     $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $< ;
1114     $(RM) $@ ;
1115     $(SED) "/^domain/d" < $(<F).po > $@ ;
1116     $(RM) $(<F).po

1118 #
1119 # When using xgettext, we want messages to go to the default domain,
1120 # rather than the specified one. This special version of the
1121 # COMPILER.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1122 # causing xgettext to put all messages into the default domain.
1123 #
1124 CPPFORPO=$(COMPILE.cpp:"$(TEXT_DOMAIN)"=TEXT_DOMAIN)

1126 .c.i:
1127     $(CPPFORPO) $< > $@

1129 .h.i:
1130     $(CPPFORPO) $< > $@

1132 .y.i:
1133     $(YACC) -d $<
1134     $(CPPFORPO) y.tab.c > $@
1135     $(RM) y.tab.c

1137 .l.i:

```

new/usr/src/Makefile.master

19

```
1138      $(LEX) $<
1139      $(CPPFORPO) lex.yy.c > $@
1140      $(RM) lex.yy.c

1142 .c.po:
1143      $(CPPFORPO) $< > $<.i
1144      $(BUILD.po)

1146 .cc.po:
1147      $(CPPFORPO) $< > $<.i
1148      $(BUILD.po)

1150 .y.po:
1151      $(YACC) -d $<
1152      $(CPPFORPO) y.tab.c > $<.i
1153      $(BUILD.po)
1154      $(RM) y.tab.c

1156 .l.po:
1157      $(LEX) $<
1158      $(CPPFORPO) lex.yy.c > $<.i
1159      $(BUILD.po)
1160      $(RM) lex.yy.c

1162 #
1163 # Rules to perform stylistic checks
1164 #
1165 .SUFFIXES: .x .xml .check .xmlchk

1167 .h.check:
1168      $(DOT_H_CHECK)

1170 .x.check:
1171      $(DOT_X_CHECK)

1173 .xml.xmlchk:
1174      $(MANIFEST_CHECK)

1176 #
1177 # Include rules to render automated sccs get rules "safe".
1178 #
1179 include $(SRC)/Makefile.noget
```

new/usr/src/Makefile.master.64

1

```
*****
2199 Wed Oct 3 20:38:51 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 #

26 # rebind basic build macros to 64-bit versions

28 CFLAGS= $(CFLAGS64)
29 CCFLAGS= $(CCFLAGS64)
30 COPTFLAG= $(COPTFLAG64)
31 CCOPTFLAG= $(CCOPTFLAG64)
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)

41 CTF_FLAGS= $(CTF_FLAGS_64)

43 OFFSETS_CREATE= $(OFFSETS_CREATE64)

45 #
46 # Moved these up to the top level here, so they can be overridden
47 #
48 LDLIBS.cmd = $(LDLIBS64)
49 LDLIBS.lib = $(LDLIBS64)

51 #
52 # Override Makefile.master's settings of GCCLIBDIR
53 #
54 GCCLIBDIR= $(GCCLIBDIR64)

56 #
57 # Override Makefile.master's settings of CC, CCC, AS and LINT
```

new/usr/src/Makefile.master.64

2

```
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

66 REAL_CC= $(CW_CC_CMD:sh)
67 REAL_CCC= $(CW_CCC_CMD:sh)

66 BUILD.SO= $(CC) $(CFLAGS) -o $@ $(GSHARED) $(DYNFLAGS) \
67 $(PICS) $(EXTPICS) -L $(ROOTLIBDIR64) $(LDLIBS)

69 #
70 # ld(1) requires the -64 option to create a 64-bit filter solely from a mapfile
71 #
72 MAPFILECLASS= -64

74 CCNEEDED = -lCrun
75 $(__GNU64)CCNEEDED = -L$(GCCLIBDIR) -lstc++
76 $(__GNU64)CCNEEDED += -lgcc_s

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 Wed Oct 3 20:38:52 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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/forker.h usdt/lazyprobe.h
51 #
52 fasttrap/tst.fasttrap.exe := LDLIBS += -ldtrace
53 fasttrap/tst.stack.exe := LDLIBS += -ldtrace
54 #
55 sysevent/tst.post.exe := LDLIBS += -lsysevent
56 sysevent/tst.post_chan.exe := LDLIBS += -lsysevent
```

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

2

```
58 ustack/tst.bigstack.exe := COPTFLAG += -xO1
60 GCC = $(ONBLD_TOOLS)/bin/$(MACH)/cw_ -gcc
61 #
62 #
63 nfs/%.o: $(SNOOPDIR)/%.c
64 $(COMPILE.c) -o $@ $< -I$(SNOOPDIR)
65 $(POST_PROCESS_0)
66 nfs/tst.call.exe: nfs/tst.call.o nfs/${SNOOPOBJS}
67 $(LINK.c) -o $@ nfs/tst.call.o nfs/${SNOOPOBJS} $(LDLIBS) -lnsl
68 $(POST_PROCESS) ; $(STRIP_STABS)
69 $(RPCSVCDIR)/%.c: $(RPCSVCDIR)/%.x
70 $(RPCGEN) -Cc $< > $@
71 nfs/${RPCSVCobjs}: $(RPCSVCDIR)/$(RPCSVCSRCS)
72 $(COMPILE.c) -o $@ $(RPCSVCDIR)/$(RPCSVCSRCS)
73 $(POST_PROCESS_0)
74 nfs/tst.call3.exe: nfs/tst.call3.o nfs/${RPCSVCobjs}
75 $(LINK.c) -o $@ nfs/tst.call3.o nfs/${RPCSVCobjs} \
76 $(LDLIBS) -lnsl -lrpcsvc
77 $(POST_PROCESS) ; $(STRIP_STABS)
78 #
79 pid/tst.gcc.exe: pid/tst.gcc.c
80 $(GCC) -o pid/tst.gcc.exe pid/tst.gcc.c $(LDLFLAGS) $(LDLIBS)
81 $(POST_PROCESS) ; $(STRIP_STABS)
82 #
83 json/tst.usdt.o: json/usdt.h
84 #
85 json/usdt.h: json/usdt.d
86 $(DTRACE) -h -s json/usdt.d -o json/usdt.h
87 #
88 CLOBBERFILES += json/usdt.h
89 #
90 json/usdt.o: json/usdt.d json/tst.usdt.o
91 $(COMPILE.d) -o json/usdt.o -s json/usdt.d json/tst.usdt.o
92 #
93 json/tst.usdt.exe: json/tst.usdt.o json/usdt.o
94 $(LINK.c) -o json/tst.usdt.exe json/tst.usdt.o json/usdt.o $(LDLIBS)
95 $(POST_PROCESS) ; $(STRIP_STABS)
96 #
97 # Tests that use the next three programs rely on the binaries having
98 # valid CTF data.
99 #
100 uctf/tst.aouttype.exe: uctf/tst.aouttype.c
101 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.aouttype.o uctf/tst.aouttype.c
102 $(CTFCONVERT) -i -L VERSION uctf/tst.aouttype.o
103 $(LINK.c) -o uctf/tst.aouttype.exe uctf/tst.aouttype.o $(LDLIBS)
104 $(CTFMERGE) -L VERSION -o $@ uctf/tst.aouttype.o
105 $(POST_PROCESS) ; $(STRIP_STABS)
106 #
107 uctf/tst.chasestrings.exe: uctf/tst.chasestrings.c
108 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.chasestrings.o uctf/tst.chasestrin
109 $(CTFCONVERT) -i -L VERSION uctf/tst.chasestrings.o
110 $(LINK.c) -o uctf/tst.chasestrings.exe uctf/tst.chasestrings.o $(LDLIBS)
111 $(CTFMERGE) -L VERSION -o $@ uctf/tst.chasestrings.o
112 $(POST_PROCESS) ; $(STRIP_STABS)
113 #
114 uctf/tst.printtype.exe: uctf/tst.printtype.c
115 $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.printtype.o uctf/tst.printtype.c
116 $(CTFCONVERT) -i -L VERSION uctf/tst.printtype.o
117 $(LINK.c) -o uctf/tst.printtype.exe uctf/tst.printtype.o $(LDLIBS)
118 $(CTFMERGE) -L VERSION -o $@ uctf/tst.printtype.o
119 $(POST_PROCESS) ; $(STRIP_STABS)
120 #
121 # This program should never have any ctf data in it.
122 #
```

```
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
```


new/usr/src/common/mapfiles/gen/Makefile

1

```
*****
4424 Wed Oct 3 20:38:53 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 FILES= \
31 $(MACH)_cc_map.noexeglobs \
32 $(MACH)_gcc_map.noexeglobs
33 #
34 $(__GNUC)FILES= $(MACH)_gcc_map.noexeglobs
35 $(__SUNC)FILES= $(MACH)_cc_map.noexeglobs
36 $(BUILD64)FILES += \
37 $(MACH64)_cc_map.noexeglobs \
38 $(MACH64)_gcc_map.noexeglobs
39 #
40 $(__GNUC)$(BUILD64)FILES += $(MACH64)_gcc_map.noexeglobs
41 $(__SUNC)$(BUILD64)FILES += $(MACH64)_cc_map.noexeglobs
42 #
43 SYMS1= syms.1
44 SYMS2= syms.2
45 MAIN1= main.1
46 MAIN2= main.2
47 #
48 TEMPLATE1= map.noexeglobs.1.template
49 TEMPLATE2= map.noexeglobs.2.template
50 #
51 all install: $(FILES)
52 #
53 lint:
54 #
55 clean:
56 $(RM) $(SYMS1) $(SYMS2) $(MAIN1) $(MAIN2)
```

new/usr/src/common/mapfiles/gen/Makefile

2

```
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 defences 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/pkg/manifests/system-dtrace-tests.mf

1

```
*****
125867 Wed Oct 3 20:38:53 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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
55 dir path=opt/SUNWdtrt/tst/common/builtinvar
56 dir path=opt/SUNWdtrt/tst/common/cg
57 dir path=opt/SUNWdtrt/tst/common/clauses
```

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

2

```
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/$(ARCH64)/chkargs mode=0555
135 file path=opt/SUNWdtrt/bin/baddof mode=0555
136 file path=opt/SUNWdtrt/bin/badioctl mode=0555
137 file path=opt/SUNWdtrt/bin/chkargs mode=0555
138 file path=opt/SUNWdtrt/bin/dstyle mode=0555
139 file path=opt/SUNWdtrt/bin/dtest mode=0555
140 file path=opt/SUNWdtrt/bin/dtfailures mode=0555
141 file path=opt/SUNWdtrt/bin/exception.lst mode=0444
142 file path=opt/SUNWdtrt/bin/jdtrace mode=0555
143 file path=opt/SUNWdtrt/lib/java/jdtrace.jar
144 file path=opt/SUNWdtrt/tst/$(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/err.uregswritel.d mode=0444
160 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswritel.exe \
161 mode=0555
162 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite2.d mode=0444
163 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite2.exe \
164 mode=0555
165 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.d mode=0444
166 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.exe mode=0555
167 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d mode=0444
168 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d.out mode=0444
169 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.exe mode=0555
170 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.d mode=0444
171 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.exe mode=0555
172 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.d mode=0444
173 file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.exe mode=0555
174 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.d mode=0444
175 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.exe mode=0555
176 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.exe mode=0555
177 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.ksh mode=0444
178 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.uregswrite.d mode=0444
179 $(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.uregswrite.exe \
180 mode=0555
181 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/usdt/tst.tailcall.ksh \
182 mode=0444
183 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d mode=0444
184 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d.out mode=0444
185 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.exe mode=0555
186 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.d mode=0444
187 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.exe mode=0555
188 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d mode=0444
189 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d.out mode=0444

```

```

190 file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.exe mode=0555
191 $(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.trapstat.ksh \
192 mode=0444
193 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_FUNC.bad.d mode=0444
194 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_MDIM.bad.d mode=0444
195 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_NULL.bad.d mode=0444
196 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_REDEF.redef.d mode=0444
197 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.avgttoofew.d mode=0444
198 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.maxnoarg.d mode=0444
199 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.minttoofew.d mode=0444
200 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.quantizetoofew.d \
201 mode=0444
202 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.stddevtoofew.d \
203 mode=0444
204 file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.sumtoofew.d mode=0444
205 file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_AGGARG.bad.d mode=0444
206 file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_PROTO.bad.d mode=0444
207 file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_IDENT.bad.d mode=0444
208 file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_UNDEF.badaggfunc.d mode=0444
209 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badexpr.d mode=0444
210 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badkey3.d mode=0444
211 file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.noeffect.d mode=0444
212 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey1.d mode=0444
213 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey2.d mode=0444
214 file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey4.d mode=0444
215 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqbad1.d \
216 mode=0444
217 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqshort.d \
218 mode=0444
219 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASEVAL.bad.d mode=0444
220 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMTYPE.lqbad1.d mode=0444
221 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMVAL.bad.d mode=0444
222 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.d mode=0444
223 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.order.d \
224 mode=0444
225 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.d mode=0444
226 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.order.d mode=0444
227 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHSTEP.d mode=0444
228 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MISMATCH.lqbadarg.d \
229 mode=0444
230 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPLARGE.lqtoofew.d \
231 mode=0444
232 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPSMALL.bad.d mode=0444
233 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPTYPE.lqbadinc.d \
234 mode=0444
235 file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPVAL.bad.d mode=0444
236 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_AGGARG.bad.d mode=0444
237 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_PROTO.bad.d mode=0444
238 file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_SCALAR.bad.d mode=0444
239 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_ARG.lquantizetoofew.d \
240 mode=0444
241 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgnoarg.d mode=0444
242 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgttoomany.d mode=0444
243 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.counttoomany.d \
244 mode=0444
245 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizenoarg.d \
246 mode=0444
247 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizetoomany.d \
248 mode=0444
249 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxnoarg.d mode=0444
250 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxtoomany.d mode=0444
251 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.minnoarg.d mode=0444
252 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.minttoomany.d mode=0444
253 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizenoarg.d \
254 mode=0444
255 file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizetoomany.d \

```

```

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

```

```

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

```

```

388 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.modby0.d mode=0444
389 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.addmin.d mode=0444
390 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.divmin.d mode=0444
391 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muladd.d mode=0444
392 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muldiv.d mode=0444
393 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d mode=0444
394 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d.out mode=0444
395 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d mode=0444
396 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d.out mode=0444
397 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.comp narrowassign.d mode=0444
398 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.comp narrowassign.d.out \
399 mode=0444
400 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d mode=0444
401 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d.out mode=0444
402 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.nonarrow.ksh mode=0444
403 file path=opt/SUNWdtrt/tst/common/arrays/err.D_ARR_BADREF.bad.d mode=0444
404 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRBIG.toobig.d mode=0444
405 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRNULL.bad.d mode=0444
406 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRSUB.bad.d mode=0444
407 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_PROTO_TYPE.badtuple.d \
408 mode=0444
409 file path=opt/SUNWdtrt/tst/common/arrays/err.D_IDENT_UNDEF.badureg.d mode=0444
410 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic1.d mode=0444
411 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic2.d mode=0444
412 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic3.d mode=0444
413 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic4.d mode=0444
414 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic5.d mode=0444
415 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic6.d mode=0444
416 file path=opt/SUNWdtrt/tst/common/arrays/tst.uregsarray.d mode=0444
417 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupgtype.d \
418 mode=0444
419 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupdtype.d \
420 mode=0444
421 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.this.d mode=0444
422 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_ARG.badsg.d mode=0444
423 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toofew.d mode=0444
424 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toomany.d mode=0444
425 file path=opt/SUNWdtrt/tst/common/assocs/err.D_SYNTAX.errassign.d mode=0444
426 file path=opt/SUNWdtrt/tst/common/assocs/err.tupoflow.d mode=0444
427 file path=opt/SUNWdtrt/tst/common/assocs/tst.cpyarray.d mode=0444
428 file path=opt/SUNWdtrt/tst/common/assocs/tst.diffprofile.d mode=0444
429 file path=opt/SUNWdtrt/tst/common/assocs/tst.initialize.d mode=0444
430 file path=opt/SUNWdtrt/tst/common/assocs/tst.invalidref.d mode=0444
431 file path=opt/SUNWdtrt/tst/common/assocs/tst.misc.d mode=0444
432 file path=opt/SUNWdtrt/tst/common/assocs/tst.orthogonality.d mode=0444
433 file path=opt/SUNWdtrt/tst/common/assocs/tst.this.d mode=0444
434 file path=opt/SUNWdtrt/tst/common/assocs/tst.valassign.d.out mode=0444
435 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.begin.d mode=0444
436 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.tick.d mode=0444
437 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d mode=0444
438 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d.out mode=0444
439 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d mode=0444
440 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d.out mode=0444
441 file \
442 path=opt/SUNWdtrt/tst/common/bitfields/err.D_ADDROF_BITFIELD.BitfieldAddress
443 mode=0444
444 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.NegBitField.d \
445 mode=0444
446 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.ZeroBitField.d \
447 mode=0444
448 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.ExceedBaseType.d \
449 mode=0444
450 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.GreaterThan64.d \
451 mode=0444
452 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFTYPE.badtype.d \
453 mode=0444

```

```

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

```

```

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

```

```

586 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DataModel64.d.ksh mode=0444
587 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh \
588 mode=0444
589 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh.out \
590 mode=0444
591 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh \
592 mode=0444
593 file \
594 path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh.out \
595 mode=0444
596 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh \
597 mode=0444
598 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh.out \
599 mode=0444
600 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh \
601 mode=0444
602 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh.out \
603 mode=0444
604 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh \
605 mode=0444
606 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh.out \
607 mode=0444
608 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh \
609 mode=0444
610 file \
611 path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh.out \
612 mode=0444
613 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithoutW.d.ksh \
614 mode=0444
615 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationOut.d.ksh \
616 mode=0444
617 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationWithO.d.ksh \
618 mode=0444
619 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus1.d.ksh mode=0444
620 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus2.d.ksh mode=0444
621 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExtraneousProbeIds.d.ksh \
622 mode=0444
623 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName1.d.ksh \
624 mode=0444
625 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName2.d.ksh \
626 mode=0444
627 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId1.d.ksh mode=0444
628 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId2.d.ksh mode=0444
629 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId3.d.ksh mode=0444
630 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule1.d.ksh \
631 mode=0444
632 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule2.d.ksh \
633 mode=0444
634 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule3.d.ksh \
635 mode=0444
636 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule4.d.ksh \
637 mode=0444
638 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProbeIdentifier.d.ksh \
639 mode=0444
640 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider1.d.ksh \
641 mode=0444
642 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider2.d.ksh \
643 mode=0444
644 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider3.d.ksh \
645 mode=0444
646 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider4.d.ksh \
647 mode=0444
648 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc1.d.ksh \
649 mode=0444
650 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc2.d.ksh \
651 mode=0444

```

```

652 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc3.d.ksh \
653     mode=0444
654 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc4.d.ksh \
655     mode=0444
656 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc5.d.ksh \
657     mode=0444
658 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc6.d.ksh \
659     mode=0444
660 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc7.d.ksh \
661     mode=0444
662 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc8.d.ksh \
663     mode=0444
664 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc9.d.ksh \
665     mode=0444
666 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID1.d.ksh \
667     mode=0444
668 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID2.d.ksh \
669     mode=0444
670 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID3.d.ksh \
671     mode=0444
672 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID4.d.ksh \
673     mode=0444
674 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID5.d.ksh \
675     mode=0444
676 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID6.d.ksh \
677     mode=0444
678 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID7.d.ksh \
679     mode=0444
680 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule1.d.ksh \
681     mode=0444
682 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule2.d.ksh \
683     mode=0444
684 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule3.d.ksh \
685     mode=0444
686 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule4.d.ksh \
687     mode=0444
688 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule5.d.ksh \
689     mode=0444
690 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule6.d.ksh \
691     mode=0444
692 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule7.d.ksh \
693     mode=0444
694 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule8.d.ksh \
695     mode=0444
696 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName1.d.ksh \
697     mode=0444
698 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName2.d.ksh \
699     mode=0444
700 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName3.d.ksh \
701     mode=0444
702 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName4.d.ksh \
703     mode=0444
704 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName5.d.ksh \
705     mode=0444
706 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName6.d.ksh \
707     mode=0444
708 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName7.d.ksh \
709     mode=0444
710 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName8.d.ksh \
711     mode=0444
712 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName9.d.ksh \
713     mode=0444
714 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider1.d.ksh \
715     mode=0444
716 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider2.d.ksh \
717     mode=0444

```

```

718 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider3.d.ksh \
719     mode=0444
720 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider4.d.ksh \
721     mode=0444
722 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider5.d.ksh \
723     mode=0444
724 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.MultipleInvalidProbeId.d.ksh \
725     mode=0444
726 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.PreprocessorStatement.d.ksh \
727     mode=0444
728 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh mode=0444
729 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh.out mode=0444
730 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh mode=0444
731 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh.out \
732     mode=0444
733 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.UnDefineNameWithCPP.d.ksh \
734     mode=0444
735 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh \
736     mode=0444
737 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh.out \
738     mode=0444
739 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh \
740     mode=0444
741 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh.out \
742     mode=0444
743 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh \
744     mode=0444
745 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh.out \
746     mode=0444
747 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbeIdentifier.d.ksh \
748     mode=0444
749 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbesWithoutZ.d.ksh \
750     mode=0444
751 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh \
752     mode=0444
753 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh.out \
754     mode=0444
755 file path=opt/SUNWdtrt/tst/common/end/err.D_IDENT_UNDEF.timespent.d mode=0444
756 file path=opt/SUNWdtrt/tst/common/end/tst.end.d mode=0444
757 file path=opt/SUNWdtrt/tst/common/end/tst.endwithoutbegin.d mode=0444
758 file path=opt/SUNWdtrt/tst/common/end/tst.multibeginend.d mode=0444
759 file path=opt/SUNWdtrt/tst/common/end/tst.multiend.d mode=0444
760 file path=opt/SUNWdtrt/tst/common/enum/err.D_DECL_IDRED.EnumSameName.d \
761     mode=0444
762 file path=opt/SUNWdtrt/tst/common/enum/err.D_UNKNOWN.RepeatIdentifiers.d \
763     mode=0444
764 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumEquality.d mode=0444
765 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumSameValue.d mode=0444
766 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumValAssign.d mode=0444
767 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.setfromscript.d \
768     mode=0444
769 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.unsetfromscript.d \
770     mode=0444
771 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh mode=0444
772 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh.out mode=0444
773 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh mode=0444
774 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh.out mode=0444
775 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh mode=0444
776 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh.out mode=0444
777 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh mode=0444
778 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh.out mode=0444
779 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh mode=0444
780 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh.out mode=0444
781 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_BADADDR.d mode=0444
782 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_DIVZERO.d mode=0444
783 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_UNKNOWN.d mode=0444

```

```

784 file path=opt/SUNWdtrt/tst/common/error/tst.error.d mode=0444
785 file path=opt/SUNWdtrt/tst/common/error/tst.errorend.d mode=0444
786 file path=opt/SUNWdtrt/tst/common/exit/err.D_PROTO_LEN.noarg.d mode=0444
787 file path=opt/SUNWdtrt/tst/common/exit/err.exitarg1.d mode=0444
788 file path=opt/SUNWdtrt/tst/common/exit/tst.basic1.d mode=0444
789 file path=opt/SUNWdtrt/tst/common/fbtprovider/err.D_PDESC_ZERO.notreturn.d \
790 mode=0444
791 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.basic.d mode=0444
792 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionentry.d mode=0444
793 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionreturnvalue.d \
794 mode=0444
795 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.ioctlargs.d mode=0444
796 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offset.d mode=0444
797 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offsetzero.d mode=0444
798 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return.d mode=0444
799 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return0.d mode=0444
800 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.tailcall.d mode=0444
801 file path=opt/SUNWdtrt/tst/common/funcs/err.D_FUNC_UNDEF.progenyofbad1.d \
802 mode=0444
803 file path=opt/SUNWdtrt/tst/common/funcs/err.D_OP_VFPTR.badop.d mode=0444
804 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.chillbadarg.d \
805 mode=0444
806 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.copyoutbadarg.d \
807 mode=0444
808 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.mobadarg.d mode=0444
809 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.raisebadarg.d \
810 mode=0444
811 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.tolower.d mode=0444
812 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.toupper.d mode=0444
813 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.allocanoarg.d \
814 mode=0444
815 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.badbreakpoint.d \
816 mode=0444
817 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoofew.d \
818 mode=0444
819 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoomany.d \
820 mode=0444
821 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrbadarg.d \
822 mode=0444
823 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrtoofew.d \
824 mode=0444
825 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoofew.d \
826 mode=0444
827 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoomany.d \
828 mode=0444
829 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoofew.d mode=0444
830 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoomany.d mode=0444
831 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtabadarg.d mode=0444
832 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoofew.d mode=0444
833 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoomany.d mode=0444
834 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.panicbadarg.d \
835 mode=0444
836 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.progenyofbad2.d \
837 mode=0444
838 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.stopbadarg.d mode=0444
839 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolower.d mode=0444
840 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolowertoomany.d \
841 mode=0444
842 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.toupper.d mode=0444
843 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tuppertoomany.d \
844 mode=0444
845 file path=opt/SUNWdtrt/tst/common/funcs/err.D_STRINGOF_TYPE.badstringof.d \
846 mode=0444
847 file path=opt/SUNWdtrt/tst/common/funcs/err.D_VAR_UNDEF.badvar.d mode=0444
848 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca.d mode=0444
849 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca2.d mode=0444

```

```

850 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy.d mode=0444
851 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy1.d mode=0444
852 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy2.d mode=0444
853 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy3.d mode=0444
854 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy4.d mode=0444
855 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy5.d mode=0444
856 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy6.d mode=0444
857 file path=opt/SUNWdtrt/tst/common/funcs/err.badchill.d mode=0444
858 file path=opt/SUNWdtrt/tst/common/funcs/err.chillbadarg.ksh mode=0444
859 file path=opt/SUNWdtrt/tst/common/funcs/err.copyout.d mode=0444
860 file path=opt/SUNWdtrt/tst/common/funcs/err.copyoutbadaddr.ksh mode=0444
861 file path=opt/SUNWdtrt/tst/common/funcs/err.copyoutstrbadaddr.ksh mode=0444
862 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoa6badaddr.d mode=0444
863 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoabadaddr.d mode=0444
864 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadaddr.d mode=0444
865 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadarg.d mode=0444
866 file path=opt/SUNWdtrt/tst/common/funcs/tst.badfreopen.ksh mode=0444
867 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d mode=0444
868 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d.out mode=0444
869 file path=opt/SUNWdtrt/tst/common/funcs/tst.bcopy.d mode=0444
870 file path=opt/SUNWdtrt/tst/common/funcs/tst.chill.ksh mode=0444
871 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d mode=0444
872 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d.out mode=0444
873 file path=opt/SUNWdtrt/tst/common/funcs/tst.copypin.d mode=0444
874 file path=opt/SUNWdtrt/tst/common/funcs/tst.copypinto.d mode=0444
875 file path=opt/SUNWdtrt/tst/common/funcs/tst.ddi_pathname.d mode=0444
876 file path=opt/SUNWdtrt/tst/common/funcs/tst.default.d mode=0444
877 file path=opt/SUNWdtrt/tst/common/funcs/tst.freopen.ksh mode=0444
878 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh mode=0444
879 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh.out mode=0444
880 file path=opt/SUNWdtrt/tst/common/funcs/tst.hton.d mode=0444
881 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d mode=0444
882 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d.out mode=0444
883 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d mode=0444
884 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d.out mode=0444
885 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d mode=0444
886 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d.out mode=0444
887 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d mode=0444
888 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d.out mode=0444
889 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d mode=0444
890 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d.out mode=0444
891 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d mode=0444
892 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d.out mode=0444
893 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owned.d mode=0444
894 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owner.d mode=0444
895 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_type_adaptive.d mode=0444
896 file path=opt/SUNWdtrt/tst/common/funcs/tst.progenyof.d mode=0444
897 file path=opt/SUNWdtrt/tst/common/funcs/tst.rand.d mode=0444
898 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d mode=0444
899 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d.out mode=0444
900 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d mode=0444
901 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d.out mode=0444
902 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d mode=0444
903 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d.out mode=0444
904 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok.d mode=0444
905 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok.d.out mode=0444
906 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok_null.d mode=0444
907 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d mode=0444
908 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d.out mode=0444
909 file path=opt/SUNWdtrt/tst/common/funcs/tst.substrminate.d mode=0444
910 file path=opt/SUNWdtrt/tst/common/funcs/tst.substrminate.d.out mode=0444
911 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d mode=0444
912 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d.out mode=0444
913 file path=opt/SUNWdtrt/tst/common/funcs/tst.tolower.d mode=0444
914 file path=opt/SUNWdtrt/tst/common/funcs/tst.toupper.d mode=0444
915 file path=opt/SUNWdtrt/tst/common/grammar/err.D_ADDR_OF_LVAL.d mode=0444

```



```

916 file path=opt/SUNWdtrt/tst/common/grammar/err.D_EMPTY.empty.d mode=0444
917 file path=opt/SUNWdtrt/tst/common/grammar/tst.clauses.d mode=0444
918 file path=opt/SUNWdtrt/tst/common/grammar/tst.stmts.d mode=0444
919 file path=opt/SUNWdtrt/tst/common/include/tst.includefirst.ksh mode=0444
920 file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef1.d mode=0444
921 file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef2.d mode=0444
922 file path=opt/SUNWdtrt/tst/common/inline/err.D_IDENT_UNDEF.recur.d mode=0444
923 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef1.d mode=0444
924 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef2.d mode=0444
925 file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.badxlate.d \
926 mode=0444
927 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineDataAssign.d mode=0444
928 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineExpression.d mode=0444
929 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d mode=0444
930 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d mode=0444
931 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineTypedef.d mode=0444
932 file path=opt/SUNWdtrt/tst/common/inline/tst.InlineWritableAssign.d mode=0444
933 file path=opt/SUNWdtrt/tst/common/io/tst.fds.d mode=0444
934 file path=opt/SUNWdtrt/tst/common/io/tst.fds.d.out mode=0444
935 file path=opt/SUNWdtrt/tst/common/io/tst.fds.exe mode=0555
936 file path=opt/SUNWdtrt/tst/common/ip/get.ipv4remote.pl mode=0555
937 file path=opt/SUNWdtrt/tst/common/ip/get.ipv6remote.pl mode=0555
938 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh mode=0444
939 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh.out mode=0444
940 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh mode=0444
941 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh.out mode=0444
942 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh mode=0444
943 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh.out mode=0444
944 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh mode=0444
945 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh.out mode=0444
946 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh mode=0444
947 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh.out mode=0444
948 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh mode=0444
949 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh.out mode=0444
950 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh mode=0444
951 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh.out mode=0444
952 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remoteicmp.ksh mode=0444
953 file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remotetcp.ksh.out mode=0444
954 file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh mode=0444
955 file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh.out mode=0444
956 file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh mode=0444
957 file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh.out mode=0444
958 file path=opt/SUNWdtrt/tst/common/java_api/test.jar
959 file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh mode=0444
960 file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh.out mode=0444
961 file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh mode=0444
962 file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh.out mode=0444
963 file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh mode=0444
964 file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh.out mode=0444
965 file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh mode=0444
966 file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh.out mode=0444
967 file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh mode=0444
968 file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh.out mode=0444
969 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.exe mode=0555
970 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh mode=0444
971 file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh.out \
972 mode=0444
973 file path=opt/SUNWdtrt/tst/common/java_api/tst.GetAggregate.ksh mode=0444
974 file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh mode=0444
975 file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh.out mode=0444
976 file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh mode=0444
977 file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh.out \
978 mode=0444
979 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.exe mode=0555
980 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh mode=0444
981 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh.out mode=0444

```

```

982 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh mode=0444
983 file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh.out \
984 mode=0444
985 file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh mode=0444
986 file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh.out mode=0444
987 file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh mode=0444
988 file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh.out mode=0444
989 file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d mode=0444
990 file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d.out mode=0444
991 file path=opt/SUNWdtrt/tst/common/json/tst.general.d mode=0444
992 file path=opt/SUNWdtrt/tst/common/json/tst.general.d.out mode=0444
993 file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d mode=0444
994 file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d.out mode=0444
995 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d mode=0444
996 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d.out mode=0444
997 file path=opt/SUNWdtrt/tst/common/json/tst.usdt.exe mode=0555
998 file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NL.char.d mode=0444
999 file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NUL.char.d mode=0444
1000 file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_DIGIT.InvalidDigit.d \
1001 mode=0444
1002 file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_OFLOW.BigInt.d mode=0444
1003 file path=opt/SUNWdtrt/tst/common/lexer/err.D_STR_NL.string.d mode=0444
1004 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack1.d mode=0444
1005 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack2.d mode=0444
1006 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack1.d mode=0444
1007 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack2.d mode=0444
1008 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack3.d mode=0444
1009 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren1.d mode=0444
1010 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren2.d mode=0444
1011 file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren3.d mode=0444
1012 file path=opt/SUNWdtrt/tst/common/lexer/tst.D_MACRO_OFLOW.ParIntOvflow.d.ksh \
1013 mode=0444
1014 file \
1015 path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR EVEN.nodivide.d
1016 mode=0444
1017 file \
1018 path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR EVEN.notfactor.d
1019 mode=0444
1020 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR MATCH.d \
1021 mode=0444
1022 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR N STEPS.d \
1023 mode=0444
1024 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR SMALL.d \
1025 mode=0444
1026 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR TYPE.d \
1027 mode=0444
1028 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOR VAL.d \
1029 mode=0444
1030 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH MATCH.d \
1031 mode=0444
1032 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH TYPE.d \
1033 mode=0444
1034 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGH VAL.d mode=0444
1035 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW MATCH.d \
1036 mode=0444
1037 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW TYPE.d mode=0444
1038 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOW VAL.d mode=0444
1039 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAG RANGE.d \
1040 mode=0444
1041 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAG TOO BIG.d \
1042 mode=0444
1043 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP MATCH.d \
1044 mode=0444
1045 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP TYPE.d \
1046 mode=0444
1047 file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEP VAL.d \

```

```

1048 mode=0444
1049 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d mode=0444
1050 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d.out mode=0444
1051 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d mode=0444
1052 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d.out mode=0444
1053 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d mode=0444
1054 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d.out mode=0444
1055 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d mode=0444
1056 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d.out mode=0444
1057 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d mode=0444
1058 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d.out mode=0444
1059 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d mode=0444
1060 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d.out mode=0444
1061 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d mode=0444
1062 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d.out mode=0444
1063 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d mode=0444
1064 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d.out mode=0444
1065 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d mode=0444
1066 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d.out mode=0444
1067 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d mode=0444
1068 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d.out mode=0444
1069 file path=opt/SUNWdtrt/tst/common/mbd/tst.dtracedcmd.ksh mode=0444
1070 file path=opt/SUNWdtrt/tst/common/mib/tst.icmp.ksh mode=0444
1071 file path=opt/SUNWdtrt/tst/common/mib/tst.tcp.ksh mode=0444
1072 file path=opt/SUNWdtrt/tst/common/mib/tst.udp.ksh mode=0444
1073 file path=opt/SUNWdtrt/tst/common/misc/err.D_PRAGMA_OPTSET.d mode=0444
1074 file path=opt/SUNWdtrt/tst/common/misc/tst.badopt.d mode=0444
1075 file path=opt/SUNWdtrt/tst/common/misc/tst.boolopt.d mode=0444
1076 file path=opt/SUNWdtrt/tst/common/misc/tst.boolopt.d.out mode=0444
1077 file path=opt/SUNWdtrt/tst/common/misc/tst.dofmax.ksh mode=0444
1078 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d mode=0444
1079 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d.out mode=0444
1080 file path=opt/SUNWdtrt/tst/common/misc/tst.enabletrace.ksh mode=0444
1081 file path=opt/SUNWdtrt/tst/common/misc/tst.haslam.d mode=0444
1082 file path=opt/SUNWdtrt/tst/common/misc/tst.include.ksh mode=0444
1083 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh mode=0444
1084 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh.out mode=0444
1085 file path=opt/SUNWdtrt/tst/common/misc/tst.roch.d mode=0444
1086 file path=opt/SUNWdtrt/tst/common/misc/tst.schrock.ksh mode=0444
1087 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGKEY.d mode=0444
1088 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGPROTO.d mode=0444
1089 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d mode=0444
1090 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d.out mode=0444
1091 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d mode=0444
1092 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d.out mode=0444
1093 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d mode=0444
1094 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d.out mode=0444
1095 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d mode=0444
1096 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d.out mode=0444
1097 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d mode=0444
1098 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d.out mode=0444
1099 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d mode=0444
1100 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d.out mode=0444
1101 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d mode=0444
1102 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d.out mode=0444
1103 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d mode=0444
1104 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d.out mode=0444
1105 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.d mode=0444
1106 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.exe mode=0555
1107 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.d mode=0444
1108 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.exe mode=0555
1109 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_BITFIELD.bitfield.d \
1110 mode=0444
1111 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.badtype.d \
1112 mode=0444
1113 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.notsou.d \

```

```

1114 mode=0444
1115 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.OffsetofNULL.d \
1116 mode=0444
1117 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.badmemb.d mode=0444
1118 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofAlias.d mode=0444
1119 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofArith.d mode=0444
1120 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofUnion.d mode=0444
1121 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d mode=0444
1122 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d.out mode=0444
1123 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d mode=0444
1124 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d.out mode=0444
1125 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d mode=0444
1126 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d.out mode=0444
1127 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.d mode=0444
1128 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.exe mode=0555
1129 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badproc.d mode=0444
1130 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_BADPID.badproc2.d mode=0444
1131 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.d mode=0444
1132 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.exe mode=0555
1133 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.d mode=0444
1134 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.exe mode=0555
1135 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.d mode=0444
1136 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.exe mode=0555
1137 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.d mode=0444
1138 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.exe mode=0555
1139 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.d mode=0444
1140 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.exe mode=0555
1141 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.d mode=0444
1142 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.exe mode=0555
1143 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.d mode=0444
1144 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.exe mode=0555
1145 file path=opt/SUNWdtrt/tst/common/pid/tst.addprobes.ksh mode=0444
1146 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.d mode=0444
1147 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.exe mode=0555
1148 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.d mode=0444
1149 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.exe mode=0555
1150 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d mode=0444
1151 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d.out mode=0444
1152 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.exe mode=0555
1153 file path=opt/SUNWdtrt/tst/common/pid/tst.float.d mode=0444
1154 file path=opt/SUNWdtrt/tst/common/pid/tst.float.exe mode=0555
1155 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.d mode=0444
1156 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.exe mode=0555
1157 file path=opt/SUNWdtrt/tst/common/pid/tst.gcc.d mode=0444
1158 file path=opt/SUNWdtrt/tst/common/pid/tst.gcc.exe mode=0555
1159 file path=opt/SUNWdtrt/tst/common/pid/tst.killonerror.ksh mode=0444
1160 file path=opt/SUNWdtrt/tst/common/pid/tst.main.ksh mode=0444
1161 file path=opt/SUNWdtrt/tst/common/pid/tst.manypids.ksh mode=0444
1162 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh mode=0444
1163 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh.out mode=0444
1164 file path=opt/SUNWdtrt/tst/common/pid/tst.problemod.ksh mode=0444
1165 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex1.ksh mode=0444
1166 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh mode=0444
1167 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh.out mode=0444
1168 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh mode=0444
1169 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh.out mode=0444
1170 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh mode=0444
1171 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh.out mode=0444
1172 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.d mode=0444
1173 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.exe mode=0555
1174 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.d mode=0444
1175 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.exe mode=0555
1176 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh mode=0444
1177 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh.out mode=0444
1178 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh.out mode=0444
1179 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh.out mode=0444

```

```

1178 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.d mode=0444
1179 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.exe mode=0555
1180 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.d mode=0444
1181 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.exe mode=0555
1182 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.d mode=0444
1183 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.exe mode=0555
1184 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.d mode=0444
1185 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.exe mode=0555
1186 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.d mode=0444
1187 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.exe mode=0555
1188 file path=opt/SUNWdtrt/tst/common/pointers/err.BadAlign.d mode=0444
1189 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.ArrayVar.d \
1190 mode=0444
1191 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.DynamicVar.d \
1192 mode=0444
1193 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.agg.d mode=0444
1194 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_NONPTR.noptr.d \
1195 mode=0444
1196 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_VOID.VoidPointerDeref.d \
1197 mode=0444
1198 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_ARRFUN.ArrayAssignment.d \
1199 mode=0444
1200 file \
1201 path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_INCOMPAT.VoidPointerArith.d \
1202 mode=0444
1203 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_LVAL.AddressChange.d \
1204 mode=0444
1205 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.NonPointerAccess.d \
1206 mode=0444
1207 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.badpointer.d mode=0444
1208 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.BadPointerAccess.d \
1209 mode=0444
1210 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.badpointer.d mode=0444
1211 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress1.d mode=0444
1212 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress2.d mode=0444
1213 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress3.d mode=0444
1214 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress4.d mode=0444
1215 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress5.d mode=0444
1216 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer1.d mode=0444
1217 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer2.d mode=0444
1218 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer3.d mode=0444
1219 file path=opt/SUNWdtrt/tst/common/pointers/tst.GlobalVar.d mode=0444
1220 file path=opt/SUNWdtrt/tst/common/pointers/tst.IntegerArithmetic1.d mode=0444
1221 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic1.d mode=0444
1222 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic2.d mode=0444
1223 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic3.d mode=0444
1224 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerAssignment.d mode=0444
1225 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer1.d mode=0444
1226 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer2.d mode=0444
1227 file path=opt/SUNWdtrt/tst/common/pointers/tst.VoidCast.d mode=0444
1228 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast1.d mode=0444
1229 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast2.d mode=0444
1230 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic1.d mode=0444
1231 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic2.d mode=0444
1232 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGERR.d mode=0444
1233 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_DEPEND.main.d mode=0444
1234 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_INVALID.d mode=0444
1235 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_MALFORM.d mode=0444
1236 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_UNUSED.UnusedPragma.d \
1237 mode=0444
1238 file path=opt/SUNWdtrt/tst/common/pragma/err.circlibdep.ksh mode=0444
1239 file path=opt/SUNWdtrt/tst/common/pragma/err.invalidlibdep.ksh mode=0444
1240 file path=opt/SUNWdtrt/tst/common/pragma/tst.libchain.ksh mode=0444
1241 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdep.ksh mode=0444
1242 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepfullyconnected.ksh \
1243 mode=0444

```

```

1244 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepsemdir.ksh mode=0444
1245 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal.ksh mode=0444
1246 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal2.ksh mode=0444
1247 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal3.d mode=0444
1248 file path=opt/SUNWdtrt/tst/common/predicates/err.D_PRED_SCALAR.NonScalarPred.d \
1249 mode=0444
1250 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.invalid.d mode=0444
1251 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.operr.d mode=0444
1252 file path=opt/SUNWdtrt/tst/common/predicates/tst.argsnotcached.d mode=0444
1253 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d mode=0444
1254 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d.out mode=0444
1255 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d mode=0444
1256 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d.out mode=0444
1257 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_IDENT_UNDEF.afterprobe.d \
1258 mode=0444
1259 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_PRAGCTL_INVALID.tabdefine.d \
1260 mode=0444
1261 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_SYNTAX.withoutpound.d \
1262 mode=0444
1263 file path=opt/SUNWdtrt/tst/common/preprocessor/err.defincomp.d mode=0444
1264 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefelsenotendif.d \
1265 mode=0444
1266 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefincomp.d mode=0444
1267 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefnotendif.d mode=0444
1268 file path=opt/SUNWdtrt/tst/common/preprocessor/err.incompelse.d mode=0444
1269 file path=opt/SUNWdtrt/tst/common/preprocessor/err.mulelse.d mode=0444
1270 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d mode=0444
1271 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d.out mode=0444
1272 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d mode=0444
1273 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d.out mode=0444
1274 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d mode=0444
1275 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d.out mode=0444
1276 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d mode=0444
1277 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d.out mode=0444
1278 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d mode=0444
1279 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d.out \
1280 mode=0444
1281 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d mode=0444
1282 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d.out mode=0444
1283 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d mode=0444
1284 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d.out mode=0444
1285 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d mode=0444
1286 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d.out mode=0444
1287 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d mode=0444
1288 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d.out mode=0444
1289 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.predicatedeclare.d \
1290 mode=0444
1291 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d mode=0444
1292 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d.out mode=0444
1293 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d mode=0444
1294 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d.out mode=0444
1295 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d mode=0444
1296 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d.out mode=0444
1297 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d mode=0444
1298 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d.out \
1299 mode=0444
1300 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.withinprobe.d mode=0444
1301 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_AGG.bad.d mode=0444
1302 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_VOID.bad.d mode=0444
1303 file path=opt/SUNWdtrt/tst/common/print/err.D_PROTO_LEN.bad.d mode=0444
1304 file path=opt/SUNWdtrt/tst/common/print/tst.array.d mode=0444
1305 file path=opt/SUNWdtrt/tst/common/print/tst.array.d.out mode=0444
1306 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d mode=0444
1307 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d.out mode=0444
1308 file path=opt/SUNWdtrt/tst/common/print/tst.dyn.d mode=0444
1309 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d mode=0444

```

```

1310 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d.out mode=0444
1311 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d mode=0444
1312 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d.out mode=0444
1313 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d mode=0444
1314 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d.out mode=0444
1315 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d mode=0444
1316 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d.out mode=0444
1317 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badagg.d \
1318 mode=0444
1319 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badfmt.d \
1320 mode=0444
1321 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badval.d \
1322 mode=0444
1323 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_PROTO.bad.d mode=0444
1324 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.jstack.d \
1325 mode=0444
1326 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.stack.d \
1327 mode=0444
1328 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.ustack.d \
1329 mode=0444
1330 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d mode=0444
1331 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d.out mode=0444
1332 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d mode=0444
1333 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d.out mode=0444
1334 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d mode=0444
1335 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d.out mode=0444
1336 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d mode=0444
1337 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d.out mode=0444
1338 file path=opt/SUNWdtrt/tst/common/printa/tst.largeusersym.ksh mode=0444
1339 file path=opt/SUNWdtrt/tst/common/printa/tst.many.d mode=0444
1340 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d mode=0444
1341 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d.out mode=0444
1342 file path=opt/SUNWdtrt/tst/common/printa/tst.stack.d mode=0444
1343 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d mode=0444
1344 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d.out mode=0444
1345 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh mode=0444
1346 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh.out mode=0444
1347 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_AGG_CONV.aggfmt.d \
1348 mode=0444
1349 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.toomany.d \
1350 mode=0444
1351 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.widths.d \
1352 mode=0444
1353 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_FMT.badfmt.d \
1354 mode=0444
1355 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_PROTO.novalue.d \
1356 mode=0444
1357 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.aggarg.d \
1358 mode=0444
1359 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.recursive.d \
1360 mode=0444
1361 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.noprec.d \
1362 mode=0444
1363 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.nowidth.d \
1364 mode=0444
1365 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badprec.d \
1366 mode=0444
1367 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badwidth.d \
1368 mode=0444
1369 file path=opt/SUNWdtrt/tst/common/printf/err.D_PROTO_LEN.toofew.d mode=0444
1370 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv1.d mode=0444
1371 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv2.d mode=0444
1372 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv3.d mode=0444
1373 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d mode=0444
1374 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d.out mode=0444
1375 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d mode=0444

```

```

1376 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d.out mode=0444
1377 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d mode=0444
1378 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d.out mode=0444
1379 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d mode=0444
1380 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d.out mode=0444
1381 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d mode=0444
1382 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d.out mode=0444
1383 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d mode=0444
1384 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d.out mode=0444
1385 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh mode=0444
1386 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh.out mode=0444
1387 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh mode=0444
1388 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh.out mode=0444
1389 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d mode=0444
1390 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d.out mode=0444
1391 file path=opt/SUNWdtrt/tst/common/printf/tst.printeE.d mode=0444
1392 file path=opt/SUNWdtrt/tst/common/printf/tst.printeE.d.out mode=0444
1393 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d mode=0444
1394 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d.out mode=0444
1395 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d mode=0444
1396 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d.out mode=0444
1397 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d mode=0444
1398 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d.out mode=0444
1399 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d mode=0444
1400 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d.out mode=0444
1401 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d mode=0444
1402 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d.out mode=0444
1403 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d mode=0444
1404 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d.out mode=0444
1405 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d mode=0444
1406 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d.out mode=0444
1407 file path=opt/SUNWdtrt/tst/common/printf/tst.widths1.d mode=0444
1408 file path=opt/SUNWdtrt/tst/common/printf/tst.wfmt.d mode=0444
1409 file path=opt/SUNWdtrt/tst/common/printf/tst.wp.d.out mode=0444
1410 file path=opt/SUNWdtrt/tst/common/privs/tst.fds.ksh mode=0444
1411 file path=opt/SUNWdtrt/tst/common/privs/tst.func_access.ksh mode=0444
1412 file path=opt/SUNWdtrt/tst/common/privs/tst.getf.ksh mode=0444
1413 file path=opt/SUNWdtrt/tst/common/privs/tst.kpriv.ksh mode=0444
1414 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivdrop.ksh mode=0444
1415 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivrestrict.ksh mode=0444
1416 file path=opt/SUNWdtrt/tst/common/privs/tst.op_access.ksh mode=0444
1417 file path=opt/SUNWdtrt/tst/common/privs/tst.procppriv.ksh mode=0444
1418 file path=opt/SUNWdtrt/tst/common/privs/tst.providers.ksh mode=0444
1419 file path=opt/SUNWdtrt/tst/common/privs/tst.tick.ksh mode=0444
1420 file path=opt/SUNWdtrt/tst/common/privs/tst.unpriv_funcs.ksh mode=0444
1421 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probeqtn.d mode=0444
1422 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probestar.d \
1423 mode=0444
1424 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.tickstar.d mode=0444
1425 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.assign.d mode=0444
1426 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declare.d mode=0444
1427 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declarein.d mode=0444
1428 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.lbraces.d mode=0444
1429 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.probespec.d mode=0444
1430 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.rbraces.d mode=0444
1431 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.redec.d mode=0444
1432 file path=opt/SUNWdtrt/tst/common/probes/tst.basi1.d mode=0444
1433 file path=opt/SUNWdtrt/tst/common/probes/tst.check.d mode=0444
1434 file path=opt/SUNWdtrt/tst/common/probes/tst.declare.d mode=0444
1435 file path=opt/SUNWdtrt/tst/common/probes/tst.declareafter.d mode=0444
1436 file path=opt/SUNWdtrt/tst/common/probes/tst.emptyprobe.d mode=0444
1437 file path=opt/SUNWdtrt/tst/common/probes/tst.pragma.d mode=0444
1438 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaaftertab.d mode=0444
1439 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmainside.d mode=0444
1440 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaoutside.d mode=0444
1441 file path=opt/SUNWdtrt/tst/common/probes/tst.probestar.d mode=0444

```

```

1442 file path=opt/SUNWdtrt/tst/common/proc/tst.create.ksh mode=0444
1443 file path=opt/SUNWdtrt/tst/common/proc/tst.discard.ksh mode=0444
1444 file path=opt/SUNWdtrt/tst/common/proc/tst.exec.ksh mode=0444
1445 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ENOENT.ksh mode=0444
1446 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ksh mode=0444
1447 file path=opt/SUNWdtrt/tst/common/proc/tst.exitcore.ksh mode=0444
1448 file path=opt/SUNWdtrt/tst/common/proc/tst.exitexit.ksh mode=0444
1449 file path=opt/SUNWdtrt/tst/common/proc/tst.exitkilled.ksh mode=0444
1450 file path=opt/SUNWdtrt/tst/common/proc/tst.signal.ksh mode=0444
1451 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.d mode=0444
1452 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.exe mode=0555
1453 file path=opt/SUNWdtrt/tst/common/proc/tst.startexit.ksh mode=0444
1454 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZERO.profile.d \
1455 mode=0444
1456 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonens.d mode=0444
1457 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonens.d \
1458 mode=0444
1459 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonsec.d mode=0444
1460 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonusec.d \
1461 mode=0444
1462 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d mode=0444
1463 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d.out mode=0444
1464 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d mode=0444
1465 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d.out mode=0444
1466 file path=opt/SUNWdtrt/tst/common/profile-n/tst.func.ksh mode=0444
1467 file path=opt/SUNWdtrt/tst/common/profile-n/tst.mod.ksh mode=0444
1468 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d mode=0444
1469 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d.out mode=0444
1470 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d mode=0444
1471 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d.out mode=0444
1472 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1473 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1474 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d mode=0444
1475 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d.out mode=0444
1476 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d mode=0444
1477 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d.out mode=0444
1478 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1479 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1480 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d mode=0444
1481 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d.out mode=0444
1482 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1483 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1484 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d mode=0444
1485 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d.out mode=0444
1486 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1487 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1488 file path=opt/SUNWdtrt/tst/common/profile-n/tst.sym.ksh mode=0444
1489 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufunc.ksh mode=0444
1490 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.exe mode=0555
1491 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh mode=0444
1492 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh.out mode=0444
1493 file path=opt/SUNWdtrt/tst/common/profile-n/tst.umod.ksh mode=0444
1494 file path=opt/SUNWdtrt/tst/common/profile-n/tst.usym.ksh mode=0444
1495 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_INVALID.wrongdec4.d \
1496 mode=0444
1497 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.nonprofile.d \
1498 mode=0444
1499 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdecl.d \
1500 mode=0444
1501 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec2.d \
1502 mode=0444
1503 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec3.d \
1504 mode=0444
1505 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d mode=0444
1506 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d.out mode=0444
1507 file path=opt/SUNWdtrt/tst/common/providers/tst.beginexit.d mode=0444

```

```

1508 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d mode=0444
1509 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d.out mode=0444
1510 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d mode=0444
1511 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d.out mode=0444
1512 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d mode=0444
1513 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d.out mode=0444
1514 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d mode=0444
1515 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d.out mode=0444
1516 file path=opt/SUNWdtrt/tst/common/providers/tst.probenam.d mode=0444
1517 file path=opt/SUNWdtrt/tst/common/providers/tst.probenam.d.out mode=0444
1518 file path=opt/SUNWdtrt/tst/common/providers/tst.probpov.d mode=0444
1519 file path=opt/SUNWdtrt/tst/common/providers/tst.probpov.d.out mode=0444
1520 file path=opt/SUNWdtrt/tst/common/providers/tst.pro fend.d mode=0444
1521 file path=opt/SUNWdtrt/tst/common/providers/tst.pro fend.d.out mode=0444
1522 file path=opt/SUNWdtrt/tst/common/providers/tst.pro fend.d.out mode=0444
1523 file path=opt/SUNWdtrt/tst/common/providers/tst.pro fend.d.out mode=0444
1524 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d mode=0444
1525 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d.out mode=0444
1526 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d mode=0444
1527 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d.out mode=0444
1528 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.d mode=0444
1529 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.exe mode=0555
1530 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.d mode=0444
1531 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.exe mode=0555
1532 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.d mode=0444
1533 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.exe mode=0555
1534 file path=opt/SUNWdtrt/tst/common/rates/tst.aggrate.d mode=0444
1535 file path=opt/SUNWdtrt/tst/common/rates/tst.aggrate.d.out mode=0444
1536 file path=opt/SUNWdtrt/tst/common/rates/tst.statusrate.d mode=0444
1537 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d mode=0444
1538 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d.out mode=0444
1539 file path=opt/SUNWdtrt/tst/common/safety/tst.basenam.d mode=0444
1540 file path=opt/SUNWdtrt/tst/common/safety/tst.caller.d mode=0444
1541 file path=opt/SUNWdtrt/tst/common/safety/tst.cleanpath.d mode=0444
1542 file path=opt/SUNWdtrt/tst/common/safety/tst.copin.d mode=0444
1543 file path=opt/SUNWdtrt/tst/common/safety/tst.copin2.d mode=0444
1544 file path=opt/SUNWdtrt/tst/common/safety/tst.ddi_pathname.d mode=0444
1545 file path=opt/SUNWdtrt/tst/common/safety/tst.dirname.d mode=0444
1546 file path=opt/SUNWdtrt/tst/common/safety/tst.errno.d mode=0444
1547 file path=opt/SUNWdtrt/tst/common/safety/tst.execname.d mode=0444
1548 file path=opt/SUNWdtrt/tst/common/safety/tst.gid.d mode=0444
1549 file path=opt/SUNWdtrt/tst/common/safety/tst.hton.d mode=0444
1550 file path=opt/SUNWdtrt/tst/common/safety/tst.index.d mode=0444
1551 file path=opt/SUNWdtrt/tst/common/safety/tst.msgdsiz.d mode=0444
1552 file path=opt/SUNWdtrt/tst/common/safety/tst.msgsiz.d mode=0444
1553 file path=opt/SUNWdtrt/tst/common/safety/tst.null.d mode=0444
1554 file path=opt/SUNWdtrt/tst/common/safety/tst.pid.d mode=0444
1555 file path=opt/SUNWdtrt/tst/common/safety/tst.ppid.d mode=0444
1556 file path=opt/SUNWdtrt/tst/common/safety/tst.progenyof.d mode=0444
1557 file path=opt/SUNWdtrt/tst/common/safety/tst.random.d mode=0444
1558 file path=opt/SUNWdtrt/tst/common/safety/tst.rw.d mode=0444
1559 file path=opt/SUNWdtrt/tst/common/safety/tst.shortstr.d mode=0444
1560 file path=opt/SUNWdtrt/tst/common/safety/tst.stack.d mode=0444
1561 file path=opt/SUNWdtrt/tst/common/safety/tst.stackdepth.d mode=0444
1562 file path=opt/SUNWdtrt/tst/common/safety/tst.stdev.d mode=0444
1563 file path=opt/SUNWdtrt/tst/common/safety/tst.strchr.d mode=0444
1564 file path=opt/SUNWdtrt/tst/common/safety/tst.strjoin.d mode=0444
1565 file path=opt/SUNWdtrt/tst/common/safety/tst.strstr.d mode=0444
1566 file path=opt/SUNWdtrt/tst/common/safety/tst.strtok.d mode=0444
1567 file path=opt/SUNWdtrt/tst/common/safety/tst.stdev.d mode=0444
1568 file path=opt/SUNWdtrt/tst/common/safety/tst.ucaller.d mode=0444
1569 file path=opt/SUNWdtrt/tst/common/safety/tst.uid.d mode=0444
1570 file path=opt/SUNWdtrt/tst/common/safety/tst.unalign.d mode=0444
1571 file path=opt/SUNWdtrt/tst/common/safety/tst.uregs.d mode=0444
1572 file path=opt/SUNWdtrt/tst/common/safety/tst.ustack.d mode=0444
1573 file path=opt/SUNWdtrt/tst/common/safety/tst.ustackdepth.d mode=0444

```

```

1574 file path=opt/SUNWdtrt/tst/common/safety/tst.vahole.d mode=0444
1575 file path=opt/SUNWdtrt/tst/common/safety/tst.violentdeath.ksh mode=0444
1576 file path=opt/SUNWdtrt/tst/common/safety/tst.zonename.d mode=0444
1577 file path=opt/SUNWdtrt/tst/common/scalars/err.D_ARR_LOCAL.thisarray.d \
1578 mode=0444
1579 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.selfthis.d \
1580 mode=0444
1581 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.thisself.d \
1582 mode=0444
1583 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_IDRED.errval.d mode=0444
1584 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dec.err.d \
1585 mode=0444
1586 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupgtype.d \
1587 mode=0444
1588 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupltype.d \
1589 mode=0444
1590 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.duppttype.d \
1591 mode=0444
1592 file path=opt/SUNWdtrt/tst/common/scalars/err.D_SYNTAX.declare.d mode=0444
1593 file path=opt/SUNWdtrt/tst/common/scalars/err.bigglobal.d mode=0444
1594 file path=opt/SUNWdtrt/tst/common/scalars/err.biglocal.d mode=0444
1595 file path=opt/SUNWdtrt/tst/common/scalars/tst.16kglobal.d mode=0444
1596 file path=opt/SUNWdtrt/tst/common/scalars/tst.16klocal.d mode=0444
1597 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d mode=0444
1598 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d.out mode=0444
1599 file path=opt/SUNWdtrt/tst/common/scalars/tst.localvar.d mode=0444
1600 file path=opt/SUNWdtrt/tst/common/scalars/tst.misc.d mode=0444
1601 file path=opt/SUNWdtrt/tst/common/scalars/tst.self.d mode=0444
1602 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray.d mode=0444
1603 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray2.d mode=0444
1604 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfthis.d mode=0444
1605 file path=opt/SUNWdtrt/tst/common/scalars/tst.this.d mode=0444
1606 file path=opt/SUNWdtrt/tst/common/scalars/tst.thisself.d mode=0444
1607 file path=opt/SUNWdtrt/tst/common/sched/tst.enqueue.d mode=0444
1608 file path=opt/SUNWdtrt/tst/common/sched/tst.oncpu.d mode=0444
1609 file path=opt/SUNWdtrt/tst/common/sched/tst.stackdepth.d mode=0444
1610 file path=opt/SUNWdtrt/tst/common/scripting/err.D_MACRO_UNDEF.invalidargs.d \
1611 mode=0444
1612 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_LVAL.rdonly.d mode=0444
1613 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_WRITE.usepidmacro.d \
1614 mode=0444
1615 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.concat.d mode=0444
1616 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.desc.d mode=0444
1617 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.inval.d mode=0444
1618 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.pid.d mode=0444
1619 file path=opt/SUNWdtrt/tst/common/scripting/tst.D_MACRO_UNUSED.overflow.ksh \
1620 mode=0444
1621 file path=opt/SUNWdtrt/tst/common/scripting/tst.arg0.d mode=0444
1622 file path=opt/SUNWdtrt/tst/common/scripting/tst.arguments.ksh mode=0444
1623 file path=opt/SUNWdtrt/tst/common/scripting/tst.assign.d mode=0444
1624 file path=opt/SUNWdtrt/tst/common/scripting/tst.basic.d mode=0444
1625 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.d mode=0444
1626 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.ksh mode=0444
1627 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.d mode=0444
1628 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.ksh mode=0444
1629 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.d mode=0444
1630 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.ksh mode=0444
1631 file path=opt/SUNWdtrt/tst/common/scripting/tst.pgid.d mode=0444
1632 file path=opt/SUNWdtrt/tst/common/scripting/tst.pid.d mode=0444
1633 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.d mode=0444
1634 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.ksh mode=0444
1635 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.d mode=0444
1636 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.ksh mode=0444
1637 file path=opt/SUNWdtrt/tst/common/scripting/tst.quite.d mode=0444
1638 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.d mode=0444
1639 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.ksh mode=0444

```

```

1640 file path=opt/SUNWdtrt/tst/common/scripting/tst.stringmacro.ksh mode=0444
1641 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.d mode=0444
1642 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.ksh mode=0444
1643 file path=opt/SUNWdtrt/tst/common/scripting/tst.trace.d mode=0444
1644 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.d mode=0444
1645 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.ksh mode=0444
1646 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargs.d mode=0444
1647 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargs.exe mode=0555
1648 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_BADREF.SizeofAssoc.d \
1649 mode=0444
1650 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_UNDEF.UnknownSymbol.d \
1651 mode=0444
1652 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.badstruct.d \
1653 mode=0444
1654 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.d mode=0444
1655 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SYNTAX.SizeofBadType.d \
1656 mode=0444
1657 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofArray.d mode=0444
1658 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofDataTypes.d mode=0444
1659 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofExpression.d mode=0444
1660 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofNULL.d mode=0444
1661 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d mode=0444
1662 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d.out mode=0444
1663 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d mode=0444
1664 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d.out mode=0444
1665 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d mode=0444
1666 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d.out mode=0444
1667 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations1.d \
1668 mode=0444
1669 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations2.d \
1670 mode=0444
1671 file \
1672 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithBreakPo
1673 mode=0444
1674 file \
1675 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithChill.d
1676 mode=0444
1677 file \
1678 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1679 mode=0444
1680 file \
1681 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1682 mode=0444
1683 file \
1684 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithPanic.d
1685 mode=0444
1686 file \
1687 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithRaise.d
1688 mode=0444
1689 file \
1690 path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithStop.d
1691 mode=0444
1692 file path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_COMM.AggAftCommit.d \
1693 mode=0444
1694 file \
1695 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithAvg.d \
1696 mode=0444
1697 file \
1698 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithCount.d
1699 mode=0444
1700 file \
1701 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithLquant.
1702 mode=0444
1703 file \
1704 path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMax.d \
1705 mode=0444

```

```

1706 file \
1707   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMin.d \
1708   mode=0444
1709 file \
1710   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithQuant.d \
1711   mode=0444
1712 file \
1713   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithStddev.d \
1714   mode=0444
1715 file \
1716   path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithSum.d \
1717   mode=0444
1718 file \
1719   path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.CommitAftCommit.d \
1720   mode=0444
1721 file path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.DisjointCommit.d \
1722   mode=0444
1723 file \
1724   path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_DREC.CommitAftDataRec.d \
1725   mode=0444
1726 file \
1727   path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.DataRecAftCommit.d \
1728   mode=0444
1729 file \
1730   path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.ExitAfterCommit.d \
1731   mode=0444
1732 file path=opt/SUNWdtrt/tst/common/speculation/err.D_EXIT_SPEC.ExitAftSpec.d \
1733   mode=0444
1734 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_MALFORM.NspecExpr.d \
1735   mode=0444
1736 file \
1737   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.HugeNspecValue.d \
1738   mode=0444
1739 file \
1740   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.InvalidSpecSize.d \
1741   mode=0444
1742 file \
1743   path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.NegSpecSize.d \
1744   mode=0444
1745 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PROTO_LEN.SpecNoId.d \
1746   mode=0444
1747 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_COMM.SpecAftCommit.d \
1748   mode=0444
1749 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_DREC.SpecAftDataRec.d \
1750   mode=0444
1751 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_SPEC.SpecAftSpec.d \
1752   mode=0444
1753 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeBufSize.d mode=0444
1754 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeNspec.d mode=0444
1755 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeSpecSize.d mode=0444
1756 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations1.d \
1757   mode=0444
1758 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations2.d \
1759   mode=0444
1760 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitAfterDiscard.d \
1761   mode=0444
1762 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitWithZero.d mode=0444
1763 file path=opt/SUNWdtrt/tst/common/speculation/tst.DataRecAftDiscard.d \
1764   mode=0444
1765 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftCommit.d mode=0444
1766 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDataRec.d \
1767   mode=0444
1768 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDiscard.d \
1769   mode=0444
1770 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardWithZero.d mode=0444
1771 file path=opt/SUNWdtrt/tst/common/speculation/tst.ExitAftDiscard.d mode=0444

```

```

1772 file path=opt/SUNWdtrt/tst/common/speculation/tst.NoSpecBuffer.d mode=0444
1773 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations1.d \
1774   mode=0444
1775 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations2.d \
1776   mode=0444
1777 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations3.d \
1778   mode=0444
1779 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculateWithRandom.d \
1780   mode=0444
1781 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationCommit.d \
1782   mode=0444
1783 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationDiscard.d \
1784   mode=0444
1785 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationID.d mode=0444
1786 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationWithZero.d \
1787   mode=0444
1788 file path=opt/SUNWdtrt/tst/common/speculation/tst.TwoSpecBuffers.d mode=0444
1789 file path=opt/SUNWdtrt/tst/common/speculation/tst.negcommit.d mode=0444
1790 file path=opt/SUNWdtrt/tst/common/speculation/tst.negspec.d mode=0444
1791 file path=opt/SUNWdtrt/tst/common/speculation/tst.zerosize.d mode=0444
1792 file path=opt/SUNWdtrt/tst/common/stability/err.D_ATTR_MIN.MinAttributes.d \
1793   mode=0444
1794 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_PROTO.bad.d mode=0444
1795 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_SIZE.d mode=0444
1796 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_FRAMES.bad.d mode=0444
1797 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_PROTO.bad.d mode=0444
1798 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_STRSIZE.bad.d mode=0444
1799 file path=opt/SUNWdtrt/tst/common/stack/tst.default.d mode=0444
1800 file path=opt/SUNWdtrt/tst/common/stackdepth/tst.default.d mode=0444
1801 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.d mode=0444
1802 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.exe mode=0555
1803 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.d mode=0444
1804 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.exe mode=0555
1805 file path=opt/SUNWdtrt/tst/common/strlen/tst.strlen1.d mode=0444
1806 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooLarge.d mode=0444
1807 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooSmall.d mode=0444
1808 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d mode=0444
1809 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d.out mode=0444
1810 file path=opt/SUNWdtrt/tst/common/struct/err.D_ADDROF_VAR.StructPointer.d \
1811   mode=0444
1812 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon.d \
1813   mode=0444
1814 file \
1815   path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon1.d \
1816   mode=0444
1817 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.circular.d \
1818   mode=0444
1819 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order.d \
1820   mode=0444
1821 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order2.d \
1822   mode=0444
1823 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.recursive.d \
1824   mode=0444
1825 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.simple.d \
1826   mode=0444
1827 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_VOIDOBJ.baddec.d mode=0444
1828 file path=opt/SUNWdtrt/tst/common/struct/err.D_PROTO_ARG.DupStructAssoc.d \
1829   mode=0444
1830 file path=opt/SUNWdtrt/tst/common/struct/tst.StructAssoc.d mode=0444
1831 file path=opt/SUNWdtrt/tst/common/struct/tst.StructDataTypes.d mode=0444
1832 file path=opt/SUNWdtrt/tst/common/struct/tst.StructInside.d mode=0444
1833 file path=opt/SUNWdtrt/tst/common/struct/tst.classelocal.d mode=0444
1834 file path=opt/SUNWdtrt/tst/common/struct/tst.classelocal.d.out mode=0444
1835 file path=opt/SUNWdtrt/tst/common/sugar/tst.else.d mode=0444
1836 file path=opt/SUNWdtrt/tst/common/sugar/tst.if.d mode=0444
1837 file path=opt/SUNWdtrt/tst/common/sugar/tst.if2.d mode=0444

```

```

1838 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_before_after.d mode=0444
1839 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_nested.d mode=0444
1840 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon.d mode=0444
1841 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon2.d mode=0444
1842 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.d mode=0444
1843 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.exe mode=0555
1844 file path=opt/SUNWdtrt/tst/common/syscall/tst.openret.ksh mode=0444
1845 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.d mode=0444
1846 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.exe mode=0555
1847 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.d mode=0444
1848 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.exe mode=0555
1849 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZERO.tick.d mode=0444
1850 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonens.d mode=0444
1851 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonensec.d mode=0444
1852 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneus.d mode=0444
1853 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneusec.d mode=0444
1854 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickarg0.d mode=0444
1855 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d mode=0444
1856 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d.out mode=0444
1857 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d mode=0444
1858 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d.out mode=0444
1859 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickns.d mode=0444
1860 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickns.d.out mode=0444
1861 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticknsec.d mode=0444
1862 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticknsec.d.out mode=0444
1863 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d mode=0444
1864 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d.out mode=0444
1865 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d mode=0444
1866 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d.out mode=0444
1867 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickus.d mode=0444
1868 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickus.d.out mode=0444
1869 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d mode=0444
1870 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d.out mode=0444
1871 file path=opt/SUNWdtrt/tst/common/trace/err.D_PROTO_LEN.bad.d mode=0444
1872 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_AGG.bad.d mode=0444
1873 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_VOID.bad.d mode=0444
1874 file path=opt/SUNWdtrt/tst/common/trace/tst.dyn.d mode=0444
1875 file path=opt/SUNWdtrt/tst/common/trace/tst.misc.d mode=0444
1876 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d mode=0444
1877 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d.out mode=0444
1878 file path=opt/SUNWdtrt/tst/common/trace/tst.string.d mode=0444
1879 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_ARG.badsize.d mode=0444
1880 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_LEN.toofew.d mode=0444
1881 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ADDR.badaddr.d \
1882 mode=0444
1883 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ARGS.d mode=0444
1884 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_DYNSIZE.d mode=0444
1885 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.negsize.d \
1886 mode=0444
1887 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.zerosize.d \
1888 mode=0444
1889 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d mode=0444
1890 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d.out mode=0444
1891 file path=opt/SUNWdtrt/tst/common/tracemem/tst.rootvtp.d mode=0444
1892 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d mode=0444
1893 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d.out mode=0444
1894 file \
1895 path=opt/SUNWdtrt/tst/common/translators/err.D_DECL_TYPERED.BadTransDecl.d \
1896 mode=0444
1897 file \
1898 path=opt/SUNWdtrt/tst/common/translators/err.D_OP_INCOMPLETE.NonExistentInpu
1899 mode=0444
1900 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl1.d \
1901 mode=0444
1902 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl3.d \
1903 mode=0444

```

```

1904 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl4.d \
1905 mode=0444
1906 file \
1907 path=opt/SUNWdtrt/tst/common/translators/err.D_TYPE_MEMBER.NonExistentInput2
1908 mode=0444
1909 file \
1910 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_INCOMPAT.BadInputType1.
1911 mode=0444
1912 file \
1913 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_MEMB.NonExistentOutput2
1914 mode=0444
1915 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_NONE.BadTransDecl6.d \
1916 mode=0444
1917 file \
1918 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_REDECL.RepeatTransDecl.
1919 mode=0444
1920 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransDecl8.d \
1921 mode=0444
1922 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransInt.d \
1923 mode=0444
1924 file \
1925 path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.NonExistentOutput1.
1926 mode=0444
1927 file path=opt/SUNWdtrt/tst/common/translators/tst.CircularTransDecl.d \
1928 mode=0444
1929 file path=opt/SUNWdtrt/tst/common/translators/tst.EmptyTransDecl.d mode=0444
1930 file path=opt/SUNWdtrt/tst/common/translators/tst.ForwardTag.d mode=0444
1931 file path=opt/SUNWdtrt/tst/common/translators/tst.InputAliasTrans.d mode=0444
1932 file path=opt/SUNWdtrt/tst/common/translators/tst.InputIntTrans.d mode=0444
1933 file path=opt/SUNWdtrt/tst/common/translators/tst.OutputAliasTrans.d mode=0444
1934 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialDereferencing.d \
1935 mode=0444
1936 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialOutputTransDefn.d \
1937 mode=0444
1938 file path=opt/SUNWdtrt/tst/common/translators/tst.ProcModelTrans.d mode=0444
1939 file path=opt/SUNWdtrt/tst/common/translators/tst.RepeatDeclaration.d \
1940 mode=0444
1941 file path=opt/SUNWdtrt/tst/common/translators/tst.SimultaneousTranslators.d \
1942 mode=0444
1943 file path=opt/SUNWdtrt/tst/common/translators/tst.StructureAssignment.d \
1944 mode=0444
1945 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh \
1946 mode=0444
1947 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh.out \
1948 mode=0444
1949 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh \
1950 mode=0444
1951 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh.out \
1952 mode=0444
1953 file path=opt/SUNWdtrt/tst/common/translators/tst.TransNonPointer.d mode=0444
1954 file path=opt/SUNWdtrt/tst/common/translators/tst.TransOutputPointer.d \
1955 mode=0444
1956 file path=opt/SUNWdtrt/tst/common/translators/tst.TransPointer.d mode=0444
1957 file path=opt/SUNWdtrt/tst/common/translators/tst.TranslateSelf.d mode=0444
1958 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionInputTrans.d mode=0444
1959 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionOutputTrans.d mode=0444
1960 file path=opt/SUNWdtrt/tst/common/typedef/err.D_DECL_IDRED.DupTypeDef.d \
1961 mode=0444
1962 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.BadExistingTypeDef.d \
1963 mode=0444
1964 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.TypeDefInClause.d \
1965 mode=0444
1966 file path=opt/SUNWdtrt/tst/common/typedef/tst.ChainTypeDef.d mode=0444
1967 file path=opt/SUNWdtrt/tst/common/typedef/tst.TypeDefDataAssign.d mode=0444
1968 file path=opt/SUNWdtrt/tst/common/types/err.D_CAST_INVALID.badcast.d mode=0444
1969 file path=opt/SUNWdtrt/tst/common/types/err.D_CG_DYN.ResultDynType.d mode=0444

```



```

1970 file path=opt/SUNWdtrt/tst/common/types/err.D_CHR_OFLOW.charconst.d mode=0444
1971 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_BADCLASS.bad.d mode=0444
1972 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_CHARATTR.badtype3.d \
1973 mode=0444
1974 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype4.d mode=0444
1975 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype5.d mode=0444
1976 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattrl1.d mode=0444
1977 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattrl2.d mode=0444
1978 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattrl3.d mode=0444
1979 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattrl4.d mode=0444
1980 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattrl5.d mode=0444
1981 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENCONST.badeval.d mode=0444
1982 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enoflow.d mode=0444
1983 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enuflow.d mode=0444
1984 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_SCOPE.scopeop.d mode=0444
1985 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_USELESS.baddec.d mode=0444
1986 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ACT.badcond.d mode=0444
1987 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ARITH.badoperand.d mode=0444
1988 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INCOMPAT.badassign.d \
1989 mode=0444
1990 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badbitop.d mode=0444
1991 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badshift.d mode=0444
1992 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badcond.d mode=0444
1993 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badincop.d mode=0444
1994 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badlogop.d mode=0444
1995 file path=opt/SUNWdtrt/tst/common/types/err.D_PROTO_LEN.badcond1.d mode=0444
1996 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badenum.d mode=0444
1997 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badid.d mode=0444
1998 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badstruct.d mode=0444
1999 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype1.d mode=0444
2000 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype2.d mode=0444
2001 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupenum.d mode=0444
2002 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupstruct.d mode=0444
2003 file path=opt/SUNWdtrt/tst/common/types/err.D_XLATE_REDECL.ResultDynType.d \
2004 mode=0444
2005 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt1.d mode=0444
2006 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt2.d mode=0444
2007 file path=opt/SUNWdtrt/tst/common/types/tst.assignops.d mode=0444
2008 file path=opt/SUNWdtrt/tst/common/types/tst.badshiftops.d mode=0444
2009 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d mode=0444
2010 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d.out mode=0444
2011 file path=opt/SUNWdtrt/tst/common/types/tst.bitops.d mode=0444
2012 file path=opt/SUNWdtrt/tst/common/types/tst.charconstants.d mode=0444
2013 file path=opt/SUNWdtrt/tst/common/types/tst.complex.d mode=0444
2014 file path=opt/SUNWdtrt/tst/common/types/tst.condexpr.d mode=0444
2015 file path=opt/SUNWdtrt/tst/common/types/tst.const.d mode=0444
2016 file path=opt/SUNWdtrt/tst/common/types/tst.constants.d mode=0444
2017 file path=opt/SUNWdtrt/tst/common/types/tst.conv.d mode=0444
2018 file path=opt/SUNWdtrt/tst/common/types/tst.enum.d mode=0444
2019 file path=opt/SUNWdtrt/tst/common/types/tst.incop.d mode=0444
2020 file path=opt/SUNWdtrt/tst/common/types/tst.intops.d mode=0444
2021 file path=opt/SUNWdtrt/tst/common/types/tst.inttypes.d mode=0444
2022 file path=opt/SUNWdtrt/tst/common/types/tst.ptrincomp.d mode=0444
2023 file path=opt/SUNWdtrt/tst/common/types/tst.ptrops.d mode=0444
2024 file path=opt/SUNWdtrt/tst/common/types/tst.relenum.d mode=0444
2025 file path=opt/SUNWdtrt/tst/common/types/tst.relstring.d mode=0444
2026 file path=opt/SUNWdtrt/tst/common/types/tst.shiftops.d mode=0444
2027 file path=opt/SUNWdtrt/tst/common/types/tst.stringconstants.d mode=0444
2028 file path=opt/SUNWdtrt/tst/common/types/tst.struct.d mode=0444
2029 file path=opt/SUNWdtrt/tst/common/types/tst.typedef.d mode=0444
2030 file path=opt/SUNWdtrt/tst/common/types/tst.unaryop.d mode=0444
2031 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid.d mode=0444
2032 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid2.d mode=0444
2033 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid3.d mode=0444
2034 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype.ksh mode=0444
2035 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype2.ksh mode=0444

```

```

2036 file path=opt/SUNWdtrt/tst/common/uctf/err.user64mode.ksh mode=0444
2037 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.exe mode=0555
2038 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.ksh mode=0444
2039 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.exe mode=0555
2040 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh mode=0444
2041 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh.out mode=0444
2042 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.exe mode=0555
2043 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.ksh mode=0444
2044 file path=opt/SUNWdtrt/tst/common/uctf/tst.linkmap.ksh mode=0444
2045 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprint.ksh mode=0444
2046 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprintarg.ksh mode=0444
2047 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.exe mode=0555
2048 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh mode=0444
2049 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh.out mode=0444
2050 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtypetarg.ksh mode=0444
2051 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh mode=0444
2052 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh.out mode=0444
2053 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh mode=0444
2054 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh.out mode=0444
2055 file path=opt/SUNWdtrt/tst/common/union/err.D_ADDR_OF_VAR.UnionPointer.d \
2056 mode=0444
2057 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon.d \
2058 mode=0444
2059 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon1.d \
2060 mode=0444
2061 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.circular.d \
2062 mode=0444
2063 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.order.d \
2064 mode=0444
2065 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.recursive.d \
2066 mode=0444
2067 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.simple.d \
2068 mode=0444
2069 file path=opt/SUNWdtrt/tst/common/union/err.D_PROTO_ARG.DupUnionAssoc.d \
2070 mode=0444
2071 file path=opt/SUNWdtrt/tst/common/union/tst.UnionAssoc.d mode=0444
2072 file path=opt/SUNWdtrt/tst/common/union/tst.UnionDataTypes.d mode=0444
2073 file path=opt/SUNWdtrt/tst/common/union/tst.UnionInside.d mode=0444
2074 file path=opt/SUNWdtrt/tst/common/usdt/tst.andpid.ksh mode=0444
2075 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.d mode=0444
2076 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.exe mode=0555
2077 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.d mode=0444
2078 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.exe mode=0555
2079 file path=opt/SUNWdtrt/tst/common/usdt/tst.badguess.ksh mode=0444
2080 file path=opt/SUNWdtrt/tst/common/usdt/tst.corruptenv.ksh mode=0444
2081 file path=opt/SUNWdtrt/tst/common/usdt/tst.deadstub.ksh mode=0444
2082 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh mode=0444
2083 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh.out mode=0444
2084 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh mode=0444
2085 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh.out mode=0444
2086 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose3.ksh mode=0444
2087 file path=opt/SUNWdtrt/tst/common/usdt/tst.eliminate.ksh mode=0444
2088 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh mode=0444
2089 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh.out mode=0444
2090 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh mode=0444
2091 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh.out mode=0444
2092 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh mode=0444
2093 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh.out mode=0444
2094 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh mode=0444
2095 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh.out mode=0444
2096 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.exe mode=0555
2097 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.ksh mode=0444
2098 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess32.ksh mode=0444
2099 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess64.ksh mode=0444
2100 file path=opt/SUNWdtrt/tst/common/usdt/tst.header.ksh mode=0444
2101 file path=opt/SUNWdtrt/tst/common/usdt/tst.include.ksh mode=0444

```

```
2102 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe.exe mode=0555
2103 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe1.ksh mode=0444
2104 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe2.ksh mode=0444
2105 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkpriv.ksh mode=0444
2106 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkunpriv.ksh mode=0444
2107 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh mode=0444
2108 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh.out mode=0444
2109 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh mode=0444
2110 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh.out mode=0444
2111 file path=opt/SUNWdtrt/tst/common/usdt/tst.nodtrace.ksh mode=0444
2112 file path=opt/SUNWdtrt/tst/common/usdt/tst.noprobes.ksh mode=0444
2113 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreap.ksh mode=0444
2114 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreapring.ksh mode=0444
2115 file path=opt/SUNWdtrt/tst/common/usdt/tst.onlyenabled.ksh mode=0444
2116 file path=opt/SUNWdtrt/tst/common/usdt/tst.reap.ksh mode=0444
2117 file path=opt/SUNWdtrt/tst/common/usdt/tst.reeval.ksh mode=0444
2118 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh mode=0444
2119 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh.out mode=0444
2120 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh mode=0444
2121 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh.out mode=0444
2122 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh mode=0444
2123 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh.out mode=0444
2124 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh mode=0444
2125 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh.out mode=0444
2126 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.d mode=0444
2127 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.exe mode=0555
2128 file path=opt/SUNWdtrt/tst/common/ustack/tst.depth.ksh mode=0444
2129 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.exe mode=0555
2130 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.ksh mode=0444
2131 file path=opt/SUNWdtrt/tst/common/vars/tst.gid.d mode=0444
2132 file path=opt/SUNWdtrt/tst/common/vars/tst.nullassign.d mode=0444
2133 file path=opt/SUNWdtrt/tst/common/vars/tst.ppid.d mode=0444
2134 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh mode=0444
2135 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh.out mode=0444
2136 file path=opt/SUNWdtrt/tst/common/vars/tst.uid.d mode=0444
2137 file path=opt/SUNWdtrt/tst/common/vars/tst.walltimestamp.d mode=0444
2138 file path=opt/SUNWdtrt/tst/common/version/tst.1.0.d mode=0444
2139 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.basic.ksh mode=0444
2140 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.hvmenable.ksh mode=0444
2141 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.memenable.ksh mode=0444
2142 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedargs.ksh mode=0444
2143 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedenable.ksh \
2144     mode=0444
2145 legacy pkg=SUNWdtrt category=internal \
2146     desc="DTrace Test Suite Internal Distribution" \
2147     hotline="Contact the DTrace discussion forum" name="DTrace Test Suite"
2148 license cr_Sun license=cr_Sun
2149 license lic_CDDL license=lic_CDDL
2150 depend fmri=runtime/java type=require
2151 depend fmri=runtime/java/runtime64 type=require
```

new/usr/src/test/os-tests/tests/file-locking/Makefile 1

```
*****
1575 Wed Oct 3 20:38:54 2018
new/usr/src/test/os-tests/tests/file-locking/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: Andy Fiddaman <omnios@citrus-it.net>
*****
```

```
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 include $(SRC)/cmd/Makefile.cmd
16 include $(SRC)/test/Makefile.com
17
18
19 OBJECTS=util.o runtests.32.o runtests.64.o acquire-lock.32.o acquire-lock.64.o
20 UTILS = util.c
21
22 PROGS = \
23     runtests \
24     acquire-lock
25
26 CSTD = $(CSTD_GNU99)
27
28 SRCS = $(PROGS:%=%.c) $(UTILS:%.o=%.c)
29 SRCS = $(PROGS:%=%.c) $(UTILS)
30 PROGS32 = $(PROGS:%=%.32)
31 PROGS64 = $(PROGS:%=%.64)
32
33 LINTS = $(PROGS:%=%.ln)
34 LINTFLAGS += -erroff=E_NAME_DEF_NOT_USED2
35 LINTFLAGS += -erroff=E_NAME_USED_NOT_DEF2
36
37 ROOTOPTDIR = $(ROOT)/opt/os-tests/tests/file-locking
38 ROOTOPTPROGS = $(PROGS32:%=$(ROOTOPTDIR)/%) \
39     $(PROGS64:%=$(ROOTOPTDIR)/%)
40
41 .KEEP_STATE:
42
43 install: $(ROOTOPTPROGS)
44
45 all: $(PROGS32) $(PROGS64)
46
47 lint: $(LINTS)
48
49 clean:
50     -rm $(PROGS32) $(PROGS64)
51
52 $(ROOTOPTPROGS): $(PROGS32) $(PROGS64) $(ROOTOPTDIR)
53
54 $(ROOTOPTDIR):
55     $(INS.dir)
56
57 $(ROOTOPTDIR)/%: %
```

new/usr/src/test/os-tests/tests/file-locking/Makefile 2

```
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
73 %.64: %.c
74     $(LINK64.c) -o $@ $< $(UTILS) $(LDLIBS64)
75     $(POST_PROCESS)
76
77
78 %.32: %.32.o util.32.o
79     $(LINK.c) -o $@ $< util.32.o $(LDLIBS)
80
81 %.32: %.c
82     $(LINK.c) -o $@ $< $(UTILS) $(LDLIBS)
83     $(POST_PROCESS)
84
85 clobber:
86     $(RM) $(PROGS32) $(PROGS64)
87
88 FRC:
```

new/usr/src/tools/cw/Makefile

1

```
*****
1990 Wed Oct 3 20:38:54 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 # Copyright 2018 Joyent, Inc.

27 PROG = cw

29 MAN1ONBLDFILES= cw.lonbld

31 include ../Makefile.tools

33 # Bootstrap problem -- we have to build cw before we can use it
34 i386_CC= $(SPRO_VROOT)/bin/cc
35 sparc_CC= $(SPRO_VROOT)/bin/cc
36 $(__GNUC)i386_CC= $(GNUC_ROOT)/bin/gcc
37 $(__GNUC)sparc_CC= $(GNUC_ROOT)/bin/gcc

39 CFLAGS += $(CCVERBOSE)

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

48 $(__GNUC)LDLIBS += -lc
49 $(__GNUC)LDFLAGS= $(MAPFILE.NES:%=-Wl,-M%)

50 CPPFLAGS += -DDEFAULT_CC_DIR='$(SPRO_VROOT)/bin'
51 CPPFLAGS += -DDEFAULT_GCC_DIR='$(GNUC_ROOT)/bin'
52 CPPFLAGS += -DDEFAULT_CPLUSPLUS_DIR='$(SPRO_VROOT)/bin'
53 CPPFLAGS += -DDEFAULT_GPLUSPLUS_DIR='$(GNUC_ROOT)/bin'
```

new/usr/src/tools/cw/Makefile

2

```
51 $(ROOTONBLDMAN1ONBLDFILES) := FILEMODE= 644

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

56 .KEEP_STATE:

58 all: $(PROG) $(MAN1ONBLDFILES)

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

62 lint: lint_PROG

64 clean:

66 #
67 # Not run by default: bootstrap...
68 check:
69 $(ROOTONBLDBINMACH)/mandoc -Tlint -Wwarning $(MAN1ONBLDFILES)

71 include ../Makefile.targ
```

```

*****
6285 Wed Oct 3 20:38:55 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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 \. Copyright 2018 Joyent, Inc.
25 \.
26 .Dd September 4, 2018
27 .Dt CW IONBLD
28 .Os
29 .Sh NAME
30 .Nm cw
31 .Nd invoke one or more compilers with argument translation
32 .Sh SYNOPSIS
33 .Nm cw
34 .Op Fl C
35 .Op Fl -versions
36 .Op Fl -noecho
37 .Fl -primary Ar compiler
38 .Op Fl -shadow Ar compiler ...
39 .Fl -
40 .Ar compiler args ...
41 .Sh DESCRIPTION
42 .Nm cw
43 is a facility for invoking one or more compilers, providing translation from
44 Sun style arguments as appropriate.
45 This allows the use of arbitrary compilers without the need to alter large
46 numbers of makefiles.
47 A mode called shadow compilation invokes multiple compilers so that warnings
48 and errors may be obtained from all of them.
49 See
50 .Sx SHADOW COMPILATION
51 for details.
52 This version of cw supports compilers with both Sun Studio 12 and GCC-style
53 command lines.
54 .Sh ARGUMENTS
55 Both the
56 .Fl -primary
57 and

```

```

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

```

```

124 may use the flag
125 .Fl _CC=
126 and the style of
127 .Em gnu
128 may use the flag
129 .Fl _g++= .
130 .El
131 .Sh SHADOW COMPILATION
132 If
133 .Fl -shadow
134 compilers are specified
135 .Nm cw
136 will invoke each shadow compiler, with the outputs modified (as well as any
137 translation for compiler style) as follows:
138 .Bl -enum
139 .It
140 If neither of
141 .Fl c ,
142 .Fl S
143 appears in the argument list (that is, linking is attempted or only the
144 pre-processor is invoked), the shadow compilers will not be invoked.
145 .It
146 If the
147 .Fl o Ar filename
148 option was provided, with or without a separating space, it will be replaced wit
149 .Fl o Ar tempfile
150 .It
151 If the option
152 .Fl o
153 was not provided,
154 .Fl o Ar tempfile
155 will be added to the end of the argument list used to invoke
156 the shadow compilers.
157 .El
158 When shadow compilation is in effect,
159 .Nm cw
160 writes to standard error each compiler's standard error output following its
161 argument list.
162 Messages from the compilers will not be interleaved.
163 If
164 .Nm cw
165 .TH CW 1ONBLD "August 16, 2018"
166 .SH NAME
167 .I cw
168 \- invoke one or more compilers with argument translation
169 .SH SYNOPSIS
170 \fBcw {-_cc|_-gcc|_-CC|_-g++} [options] [compiler args...]\fP
171 .SH DESCRIPTION
172 .LP
173 .I cw
174 is a facility for invoking one or more compilers, providing
175 translation from Sun Studio arguments as appropriate. This allows
176 the use of arbitrary compilers without the need to alter large
177 numbers of makefiles. A mode called shadow compilation, the default,
178 invokes two different compilers so that warnings and errors may be
179 obtained from both. See SHADOW COMPILATION for details. This version
180 of cw supports Sun Studio 10 and 11 and gcc 3.4.3 as shipped in
181 Solaris.
182 .SH ARGUMENTS
183 .LP
184 Exactly one of the following arguments is required, and must appear
185 immediately following \fBcw\fP in the argument list:
186 .TP 4
187 .B \-_cc
188 Select Sun Studio C (cc) as the primary compiler.
189 .TP 4

```

```

190 .B \-_gcc
191 Select GNU C (gcc) as the primary compiler.
192 .TP 4
193 .B \-_CC
194 Select Sun Studio C++ (CC) as the primary compiler.
195 .TP 4
196 .B \-_g++
197 Select GNU C++ (g++) as the primary compiler.
198 .SH OPTIONS
199 .LP
200 The following options are supported:
201 .TP 4
202 .B \-_compiler
203 Do not invoke any compiler; instead, write to standard output the
204 full path to the primary compiler that would be invoked. If this
205 option is given, it must appear immediately following the compiler
206 selection argument, and all other options will be disregarded.
207 .TP 4
208 .B \-_versions
209 Print the cw version, followed in turn by the path to and versions
210 of each compiler which would be invoked. If this option is given,
211 it must appear immediately following the compiler selection argument,
212 and all other options will be disregarded.
213 .TP 4
214 .B \-_noecho
215 Normally, cw writes to standard error the actual compiler commands
216 invoked, each prefixed with '+'. This option suppresses these
217 messages.
218 .TP 4
219 .B \-_cc=, \-_gcc=, \-_CC=, \-_g++=
220 Pass compiler-dependent options. See ARGUMENT TRANSLATION.
221 .SH SHADOW COMPILATION
222 .LP
223 Unless otherwise directed by environment variables (see ENVIRONMENT),
224 \fBcw\fP will invoke both the compiler specified by its first argument
225 (primary) and a designated alternate compiler (shadow). The primary
226 compiler will be invoked with the arguments given to cw, translated
227 as described in ARGUMENT TRANSLATION. The secondary compiler will be
228 invoked in parallel with the primary (but see ENVIRONMENT); its
229 arguments will likewise be translated. However, the shadow compiler's
230 arguments will also be modified as follows:
231 .TP 4
232 .B 1.
233 If none of \fI-c\fP, \fI-E\fP, \fI-P\fP, or \fI-S\fP appears in the
234 argument list (that is, linking is attempted), the shadow compiler
235 will not be invoked. This is because the objects built with that
236 compiler which would be linked have been previously discarded.
237 .TP 4
238 .B 2.
239 If an option of the form \fI-ofilename\fP was provided, it will be
240 replaced by two options of the form \fI-o\fP \fItempfile\fP.
241 .TP 4
242 .B 3.
243 If the option \fI-o\fP was provided, its following argument will be
244 replaced by \fItempfile\fP.
245 .TP 4
246 .B 4.
247 If neither of the above options was provided, two options of the
248 form \fI-o\fP \fItempfile\fP will be added to the end of the argument
249 list used to invoke the shadow compiler.
250 .LP
251 In all three cases, \fItempfile\fP is the name of a temporary file
252 which will be removed by \fBcw\fP before it terminates.
253 .LP
254 When shadow compilation is in effect, \fBcw\fP writes to standard error
255 each compiler's standard error output following its argument list.

```

```

115 Messages from the two compilers will not be interleaved. If \fBcw\fP
165 is used to invoke the preprocessor and no output location is specified,
166 .Nm cw
167 will write to standard output the primary compiler's standard output.
168 .Pp
169 Because the Sun compilers write intermediate objects to fixed
117 \fBcw\fP will write to standard output the primary compiler's
118 standard output, and the secondary compiler's standard output will
119 be discarded.
120 .LP
121 Because the Studio compilers write intermediate objects to fixed
170 filenames in the current directory when instructed to compile and
171 link multiple source files via a single command line, it would be
172 unsafe to invoke more than one compiler in this fashion.
173 Therefore
174 .Nm cw
175 does not accept multiple source files unless the preprocessor is to be
176 invoked.
177 An attempt to invoke
178 .Nm cw
179 in this manner will result in an error.
180 .Sh ARGUMENT TRANSLATION
181 If the compiler to be invoked is a GNU-style C or C++ compiler, a set of
124 unsafe to invoke more than one compiler in this fashion. Therefore
125 \fBcw\fP does not accept multiple source files unless the
126 preprocessor is to be invoked. An attempt to invoke \fBcw\fP in
127 this manner will result in an error.
128 .SH ARGUMENT TRANSLATION
129 .LP
130 The arguments provided to \fBcw\fP will be passed through to each
131 compiler invoked, altered as follows:
132 .TP 4
133 .B 1.
134 Options intended for \fBcw\fP itself are removed.
135 .TP 4
136 .B 2.
137 The shadow compiler's arguments are modified as described in
138 SHADOW COMPILATION so that its output is discarded.
139 .TP 4
140 .B 3.
141 If the compiler to be invoked is a GNU C or C++ compiler, a set of
182 default flags is added to the beginning of the argument list, and the
183 remaining arguments are translated to their closest appropriate
184 semantic equivalents and passed in the same order as their
185 counterparts given to
186 .Nm cw .
187 See the comments at the head of
188 .Pa usr/src/tools/cw/cw.c
189 for a detailed list of translations.
190 .Sh ENVIRONMENT
191 .Bl -tag -width indent
192 .It CW_SHADOW_SERIAL
193 If this variable is set in the environment, invoke the primary compiler, wait
194 for it to complete, then invoke the shadow compilers.
195 Normally the primary and shadow compilers are invoked in parallel.
196 .It CW_NO_EXEC
197 If this variable is set in the environment, write the usual output to
198 standard error but do not actually invoke any compiler.
199 This is useful for debugging the translation engine.
200 .El
201 .Sh EXIT STATUS
145 counterparts given to \fBcw\fP. Arguments which begin with
146 \fI-gcc=\fP or \fI-g++=\fP will be passed in place with the
147 leading portion removed. Arguments which begin with \fI-cc=\fP or
148 \fI-CC=\fP will be discarded. See the comments at the head of
149 \fIusr/src/tools/cw/cw.c\fP for a detailed list of translations.

```

```

150 .TP 4
151 .B 4.
152 If the compiler to be invoked is a Studio C or C++ compiler, the
153 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. This is
173 useful for debugging the translation engine.
174 .TP 4
175 .B CW_CC, CW_CPLUSPLUS
176 If these variables are set in the environment, they specify the full
177 pathname for the Studio C and C++ compilers, respectively.
178 .TP 4
179 .B CW_CC_DIR, CW_CPLUSPLUS_DIR, SPRO_VROOT, SPRO_ROOT, BUILD_TOOLS
180 If CW_CC or CW_CPLUSPLUS are not set,
181 these variables define the search path for Studio compilers as
182 follows:
183 .B
184 If CW_CC_DIR is set, the Studio C compiler in \fI$CW_CC_DIR\fP
185 will be used. Likewise, if CW_CPLUSPLUS_DIR is set, the Studio C++
186 compiler in \fI$CW_CPLUSPLUS_DIR\fP will be used.
187 Otherwise, if SPRO_VROOT is set, the Studio C and C++ compilers
188 in \fI$SPRO_VROOT/bin\fP will be used.
189 Otherwise, if SPRO_ROOT is set, the Studio C and C++ compilers
190 in \fI$SPRO_ROOT/SS12/bin\fP will be used.
191 Otherwise, if BUILD_TOOLS is set, the Studio C and C++ compilers
192 in \fI$BUILD_TOOLS/SUNWspr0/SS12/bin\fP will be used.
193 Otherwise, the Studio compilers in a predefined default location
194 will be used.
195 .TP 4
196 .B CW_GCC, CW_GPLUSPLUS
197 If these variables are set in the environment, they specify the full
198 pathname for the GNU C and C++ compilers, respectively.
199 .TP 4
200 .B CW_GCC_DIR, CW_GPLUSPLUS_DIR, GNUC_ROOT
201 If CW_GCC or CW_GPLUSPLUS are not set,
202 these variables alter the search path for GNU compilers in a manner
203 similar to that described above for the Studio compilers. Specifically:
204 .LP
205 If CW_GCC_DIR is set, the GNU C compiler in \fI$CW_GCC_DIR\fP
206 will be used. Likewise, if CW_GPLUSPLUS_DIR is set, the GNU C++
207 compiler in \fI$CW_GPLUSPLUS_DIR\fP will be used.
208 Otherwise, if GNUC_ROOT is set, the GNU C and C++ compilers
209 in \fI$GNUC_ROOT/bin\fP will be used. Previously \fI$GCC_ROOT\fP was
210 used to in place of \fI$GNUC_ROOT\fP; however, because that environment
211 variable has special meaning to gcc, if it is encountered in bldenv or
212 nightly(1ONBLD), it will be translated to \fI$GNUC_ROOT\fP.
213 .LP
214 If none of the aforementioned environment variables are set, the GNU
215 compilers in a predefined default location will be used.

```

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



```

*****
45749 Wed Oct 3 20:38:56 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: Andy Fiddaman <omnios@citrus-it.net>
*****

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

23 /*
24 * Copyright 2018, Richard Lowe.
25 * Copyright 2011, Richard Lowe.
26 */
27 * Copyright 2010 Sun Microsystems, Inc. All rights reserved.
28 * Use is subject to license terms.
29 */

31 /*
32 * Wrapper for the GNU C compiler to make it accept the Sun C compiler
33 * arguments where possible.
34 *
35 * Since the translation is inexact, this is something of a work-in-progress.
36 *
37 */

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

42 /*
43 * -#           Verbose mode
44 * -###        Show compiler commands built by driver, no compilation
45 * -A<name[(tokens)]> Preprocessor predicate assertion
46 * -B[static|dynamic]> Specify dynamic or static binding
47 * -C          Prevent preprocessor from removing comments
48 * -c          Compile only - produce .o files, suppress linking
49 * -cg92       Alias for -xtarget=ssl000
50 * -D<name=[token]> Associate name with token as if by #define
51 * -d[y|n]     dynamic [-dy] or static [-dn] option to linker
52 * -E          Compile source through preprocessor only, output to stdout
53 * -errroff=<t> Suppress warnings specified by tags t(%none, %all, <tag list>)
54 * -errtags=<a> Display messages with tags a(no, yes)
55 * -errwarn=<t> Treats warnings specified by tags t(%none, %all, <tag list>)

```

```

56 *           as errors
57 * -fast       Optimize using a selection of options
58 * -fd        Report old-style function definitions and declarations
59 * -features=zla Allow zero-length arrays
60 * -flags      Show this summary of compiler options
61 * -fnonstd    Initialize floating-point hardware to non-standard preferences
62 * -fns[=<yes|no>] Select non-standard floating point mode
63 * -fprecision=<p> Set FP rounding precision mode p(single, double, extended)
64 * -fround=<r>  Select the IEEE rounding mode in effect at startup
65 * -fsimple[=<n>] Select floating-point optimization preferences <n>
66 * -fsingle    Use single-precision arithmetic (-Xt and -Xs modes only)
67 * -ftrap=<t>  Select floating-point trapping mode in effect at startup
68 * -fstore     force floating pt. values to target precision on assignment
69 * -G          Build a dynamic shared library
70 * -g          Compile for debugging
71 * -H          Print path name of each file included during compilation
72 * -h <name>   Assign <name> to generated dynamic shared library
73 * -I<dir>     Add <dir> to preprocessor #include file search path
74 * -i          Passed to linker to ignore any LD_LIBRARY_PATH setting
75 * -keeptmp    Keep temporary files created during compilation
76 * -KPIC       Compile position independent code with 32-bit addresses
77 * -Kpic       Compile position independent code
78 * -L<dir>     Pass to linker to add <dir> to the library search path
79 * -l<name>    Link with library lib<name>.a or lib<name>.so
80 * -mc         Remove duplicate strings from .comment section of output files
81 * -mr         Remove all strings from .comment section of output files
82 * -mr "string" Remove all strings and append "string" to .comment section
83 * -mt         Specify options needed when compiling multi-threaded code
84 * -native     Find available processor, generate code accordingly
85 * -nofstore   Do not force floating pt. values to target precision
86 *             on assignment
87 * -nolib      Same as -xnolib
88 * -noqueue    Disable queuing of compiler license requests
89 * -norunpath  Do not build in a runtime path for shared libraries
90 * -O          Use default optimization level (-xO2 or -xO3. Check man page.)
91 * -o <outputfile> Set name of output file to <outputfile>
92 * -P          Compile source through preprocessor only, output to .i file
93 * -PIC        Alias for -KPIC or -xcode=pic32
94 * -p          Compile for profiling with prof
95 * -pic        Alias for -Kpic or -xcode=pic13
96 * -Q[y|n]     Emit/don't emit identification info to output file
97 * -qp         Compile for profiling with prof
98 * -R<dir[:dir]> Build runtime search path list into executable
99 * -S          Compile and only generate assembly code (.s)
100 * -s          Strip symbol table from the executable file
101 * -t          Turn off duplicate symbol warnings when linking
102 * -U<name>    Delete initial definition of preprocessor symbol <name>
103 * -V          Report version number of each compilation phase
104 * -v          Do stricter semantic checking
105 * -W<c>,<arg> Pass <arg> to specified component <c> (a,l,m,p,0,2,h,i,u)
106 * -w          Suppress compiler warning messages
107 * -Xa         Compile assuming ANSI C conformance, allow K & R extensions
108 *             (default mode)
109 * -Xc         Compile assuming strict ANSI C conformance
110 * -Xs         Compile assuming (pre-ANSI) K & R C style code
111 * -Xt         Compile assuming K & R conformance, allow ANSI C
112 * -x386       Generate code for the 80386 processor
113 * -x486       Generate code for the 80486 processor
114 * -xarch=<a>  Specify target architecture instruction set
115 * -xbuiltin[=<b>] When profitable inline, or substitute intrinsic functions
116 *             for system functions, b={%all,%none}
117 * -xCC        Accept C++ style comments
118 * -xchar_byte_order=<o> Specify multi-char byte order <o> (default, high, low)
119 * -xchip=<c>  Specify the target processor for use by the optimizer
120 * -xcode=<c>  Generate different code for forming addresses
121 * -xcrossfile[=<n>] Enable optimization and inlining across source files,

```

```

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

168 /*
169 * Translation table:
170 */
171 /*
172 * -#                -v
173 * -###              error
174 * -A<name[(tokens)]> pass-thru
175 * -B<[static|dynamic]> pass-thru (syntax error for anything else)
176 * -C                pass-thru
177 * -c                pass-thru
178 * -cg92             -m32 -mcpu=v8 -mtune=supersparc (SPARC only)
179 * -D<name[=token]>  pass-thru
180 * -dy or -dn        -Wl,-dy or -Wl,-dn
181 * -E                pass-thru
182 * -erroff=E_EMPTY_TRANSLATION_UNIT ignore
183 * -errtags=%all     -Wall
184 * -errwarn=%all     -Werror else -Wno-error
185 * -fast             error
186 * -fd               error
187 * -features=zla     ignore

```

```

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

```

```

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

295 #include <ctype.h>
296 #include <err.h>
297 #include <errno.h>
298 #include <fcntl.h>
299 #include <getopt.h>
300 #include <stdio.h>
301 #include <stdlib.h>
302 #include <string.h>
303 #include <unistd.h>

305 #include <sys/param.h>
306 #include <sys/stat.h>
307 #include <sys/types.h>
296 #include <unistd.h>
297 #include <string.h>
298 #include <stdlib.h>
299 #include <ctype.h>
300 #include <fcntl.h>
301 #include <errno.h>
302 #include <stdarg.h>
308 #include <sys/utsname.h>
304 #include <sys/param.h>
305 #include <sys/isa_defs.h>
309 #include <sys/wait.h>
307 #include <sys/stat.h>

```

```

311 #define CW_F_CXX      0x01
312 #define CW_F_SHADOW  0x02
313 #define CW_F_EXEC    0x04
314 #define CW_F_ECHO    0x08
315 #define CW_F_XLATE   0x10
316 #define CW_F_PROG    0x20

316 typedef enum cw_compiler {
317     CW_C_CC = 0,
318     CW_C_GCC
319 } cw_compiler_t;

321 static const char *cmds[] = {
322     "cc", "CC",
323     "gcc", "g++"
324 };

326 static char default_dir[2][MAXPATHLEN] = {
327     DEFAULT_CC_DIR,
328     DEFAULT_GCC_DIR,
329 };

331 #define CC(ctx) \
332     (((ctx)->i_flags & CW_F_SHADOW) ? \
333     ((ctx)->i_compiler == CW_C_CC ? CW_C_CC : \
334     (ctx)->i_compiler)

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

318 typedef enum cw_op {
319     CW_O_NONE = 0,
320     CW_O_PREPROCESS,
321     CW_O_COMPILE,
322     CW_O_LINK
323 } cw_op_t;
    unchanged portion omitted

333 typedef enum {
334     GNU,
335     SUN
336 } compiler_style_t;

338 typedef struct {
339     char *c_name;
340     char *c_path;
341     compiler_style_t c_style;
342 } cw_compiler_t;

344 typedef struct cw_ictx {
345     struct cw_ictx *i_next;
346     cw_compiler_t *i_compiler;
347     cw_compiler_t i_compiler;
348     struct aelist *i_ae;
349     uint32_t i_flags;
350     int i_oldargc;
351     char **i_oldargv;
352     pid_t i_pid;
353     char i_discard[MAXPATHLEN];
354     char *i_stderr;
354 } cw_ictx_t;
    unchanged portion omitted

379 /*
380 * The translation table for the -xarch= flag used in the Studio compilers.

```

```

381 */
382 static const xarch_table_t xtbl[] = {
383 #if defined(__x86)
384     { "generic",      SS11, {NULL} },
385     { "generic64",   (SS11|M64), { "-m64", "-mtune=opteron" } },
386     { "amd64",       (SS11|M64), { "-m64", "-mtune=opteron" } },
387     { "386",         SS11, { "-march=i386" } },
388     { "pentium_pro", SS11, { "-march=pentiumpro" } },
389     { "sse",         SS11, { "-msse", "-mfpmath=sse" } },
390     { "sse2",        SS11, { "-msse2", "-mfpmath=sse" } },
391 #elif defined(__sparc)
392     { "generic",     (SS11|M32), { "-m32", "-mcpu=v8" } },
393     { "generic64",   (SS11|M64), { "-m64", "-mcpu=v9" } },
394     { "v8",          (SS11|M32), { "-m32", "-mcpu=v8", "-mno-v8plus" } },
395     { "v8plus",      (SS11|M32), { "-m32", "-mcpu=v9", "-mv8plus" } },
396     { "v8plusa",     (SS11|M32), { "-m32", "-mcpu=ultrasparc", "-mv8plus",
397     "-mvis" } },
398     { "v8plusb",     (SS11|M32), { "-m32", "-mcpu=ultrasparc3", "-mv8plus",
399     "-mvis" } },
400     { "v9",          (SS11|M64), { "-m64", "-mcpu=v9" } },
401     { "v9a",         (SS11|M64), { "-m64", "-mcpu=ultrasparc", "-mvis" } },
402     { "v9b",         (SS11|M64), { "-m64", "-mcpu=ultrasparc3", "-mvis" } },
403     { "sparc",       SS12, { "-mcpu=v9", "-mv8plus" } },
404     { "sparcvis",    SS12, { "-mcpu=ultrasparc", "-mvis" } },
405     { "sparcvis2",   SS12, { "-mcpu=ultrasparc3", "-mvis" } }
406 #endif
407 };

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

420 static const char *progname;

411 static const char *xchip_tbl[] = {
412 #if defined(__x86)
413     "386",          "-mtune=i386", NULL,
414     "486",          "-mtune=i486", NULL,
415     "pentium",      "-mtune=pentium", NULL,
416     "pentium_pro", "-mtune=pentiumpro", NULL,
417 #elif defined(__sparc)
418     "super",        "-mtune=supersparc", NULL,
419     "ultra",         "-mtune=ultrasparc", NULL,
420     "ultra3",       "-mtune=ultrasparc3", NULL,
421 #endif
422     NULL,           NULL
423 };

    unchanged_portion_omitted

453 static void
454 nomem(void)
455 {
456     errx(1, "out of memory");
457     (void) fprintf(stderr, "%s: error: out of memory\n", progname);
458     exit(1);
459 }

459 static void
472 cw_perror(const char *fmt, ...)
473 {
474     va_list ap;
475     int saved_errno = errno;

477     (void) fprintf(stderr, "%s: error: ", progname);

479     va_start(ap, fmt);
480     (void) vfprintf(stderr, fmt, ap);

```

```

481     va_end(ap);

483     (void) fprintf(stderr, "%s\n", strerror(saved_errno));
484 }

486 static void
460 newae(struct aelist *ael, const char *arg)
461 {
462     struct ae *ae;

464     if ((ae = calloc(sizeof(*ae), 1)) == NULL)
465         nomem();
466     ae->ae_arg = strdup(arg);
467     if (ael->ael_tail == NULL)
468         ael->ael_head = ae;
469     else
470         ael->ael_tail->ae_next = ae;
471     ael->ael_tail = ae;
472     ael->ael_argc++;
473 }

    unchanged_portion_omitted

488 static void
489 error(const char *arg)
490 {
491     errx(2, "error: mapping failed at or near arg '%s'", arg);
492     (void) fprintf(stderr,
493         "%s: error: mapping failed at or near arg '%s'\n", progname, arg);
494     exit(2);
495 }

    unchanged_portion_omitted

523 /* ARGSUSED */
524 static void
525 Xamode(struct aelist __unused *h)
526 Xamode(struct aelist *h)
527 {

    unchanged_portion_omitted

553 static void
554 usage()
555 {
556     extern char * progname;
557     (void) fprintf(stderr,
558         "usage: %s [-C] [--versions] --primary <compiler> "
559         "[--shadow <compiler>]... -- cflags...\n",
560         progname);
561     (void) fprintf(stderr, "compilers take the form: name,path,style\n"
562         "- name: a unique name usable in flag specifiers\n"
563         "- path: path to the compiler binary\n"
564         "- style: the style of flags expected: either sun or gnu\n");
565     (void) fprintf(stderr, "usage: %s { -_cc | -_gcc | -_CC | -_g++ } [ -_compiler | ... ]\n",
566         progname);
567     exit(2);
568 }

    unchanged_portion_omitted

615 static void
616 do_gcc(cw_ictx_t *ctx)
617 {
618     int c;
619     int pic = 0, nolIBC = 0;
620     int in_output = 0, seen_o = 0, c_files = 0;
621     cw_op_t op = CW_O_LINK;
622     char *model = NULL;

```

```

623 char *nameflag;
624 int mflag = 0;

626 if (ctx->i_flags & CW_F_PROG) {
627     newae(ctx->i_ae, "--version");
628     return;
629 }

631 newae(ctx->i_ae, "-fident");
632 newae(ctx->i_ae, "-finline");
633 newae(ctx->i_ae, "-fno-inline-functions");
634 newae(ctx->i_ae, "-fno-builtin");
635 newae(ctx->i_ae, "-fno-asm");
636 newae(ctx->i_ae, "-fdiagnostics-show-option");
637 newae(ctx->i_ae, "-nodefaultlibs");

639 #if defined(__sparc)
640 /*
641  * The SPARC ldd and std instructions require 8-byte alignment of
642  * their address operand. gcc correctly uses them only when the
643  * ABI requires 8-byte alignment; unfortunately we have a number of
644  * pieces of buggy code that doesn't conform to the ABI. This
645  * flag makes gcc work more like Studio with -xmemalign=4.
646  */
647 newae(ctx->i_ae, "-mno-integer-ldd-std");
648 #endif

650 /*
651  * This is needed because 'u' is defined
652  * under a conditional on 'sun'. Should
653  * probably just remove the conditional,
654  * or make it be dependent on '__sun'.
655  *
656  * -Dunix is also missing in enhanced ANSI mode
657  */
658 newae(ctx->i_ae, "-D_sun");

660 if (asprintf(&nameflag, "-_%s=", ctx->i_compiler->c_name) == -1)
661     nomem();

663 /*
664  * Walk the argument list, translating as we go ..
665  */

666 while (--ctx->i_oldargc > 0) {
667     char *arg = **ctx->i_oldargv;
668     size_t arglen = strlen(arg);

670     if (*arg == '-') {
671         arglen--;
672     } else {
673         /*
674          * Discard inline files that gcc doesn't grok
675          */
676         if (!in_output && arglen > 3 &&
677             strcmp(arg + arglen - 3, ".il") == 0)
678             continue;

680         if (!in_output && arglen > 2 &&
681             arg[arglen - 2] == '.' &&
682             (arg[arglen - 1] == 'S' || arg[arglen - 1] == 's' ||
683              arg[arglen - 1] == 'c' || arg[arglen - 1] == 'i'))
684             c_files++;

686     /*
687      * Otherwise, filenames and partial arguments

```

```

688     * are passed through for gcc to chew on. However,
689     * output is always discarded for the secondary
690     * compiler.
691     */
692     if ((ctx->i_flags & CW_F_SHADOW) && in_output)
693         newae(ctx->i_ae, ctx->i_discard);
694     else
695         newae(ctx->i_ae, arg);
696     in_output = 0;
697     continue;
698 }

700 if (ctx->i_flags & CW_F_CXX) {
701     if (strncmp(arg, "-g++=", 6) == 0) {
702         newae(ctx->i_ae, strchr(arg, '=') + 1);
703         continue;
704     }
705     if (strncmp(arg, "-compat=", 8) == 0) {
706         /* discard -compat=4 and -compat=5 */
707         continue;
708     }
709     if (strcmp(arg, "-Qoption") == 0) {
710         /* discard -Qoption and its two arguments */
711         if (ctx->i_oldargc < 3)
712             error(arg);
713         ctx->i_oldargc -= 2;
714         ctx->i_oldargv += 2;
715         continue;
716     }
717     if (strcmp(arg, "-xwe") == 0) {
718         /* turn warnings into errors */
719         newae(ctx->i_ae, "-Werror");
720         continue;
721     }
722     if (strcmp(arg, "-noex") == 0) {
723         /* no exceptions */
724         newae(ctx->i_ae, "-fno-exceptions");
725         /* no run time type descriptor information */
726         newae(ctx->i_ae, "-fno-rtti");
727         continue;
728     }
729     if (strcmp(arg, "-pic") == 0) {
730         newae(ctx->i_ae, "-fpic");
731         pic = 1;
732         continue;
733     }
734     if (strcmp(arg, "-PIC") == 0) {
735         newae(ctx->i_ae, "-fPIC");
736         pic = 1;
737         continue;
738     }
739     if (strcmp(arg, "-norunpath") == 0) {
740         /* gcc has no corresponding option */
741         continue;
742     }
743     if (strcmp(arg, "-nolib") == 0) {
744         /* -nodefaultlibs is on by default */
745         nolibc = 1;
746         continue;
747     }
748 #if defined(__sparc)
749     if (strcmp(arg, "-cg92") == 0) {
750         mflag |= xlate_xtb(ctx->i_ae, "v8");
751         xlate(ctx->i_ae, "super", xchip_ttbl);
752         continue;
753     }

```

```

754 #endif /* __sparc */
755 }

757     switch ((c = arg[1])) {
758     case '_':
759         if ((strcmp(arg, nameflag, strlen(nameflag)) == 0) ||
760             (strcmp(arg, "_gcc=", 6) == 0) ||
761             (strcmp(arg, "_gnu=", 6) == 0)) {
762             newae(ctx->i_ae, strchr(arg, '=') + 1);
763         }
764         if (strcmp(arg, "_noecho") == 0)
765             ctx->i_flags &= ~CW_F_ECHO;
766         else if (strcmp(arg, "_cc=", 5) == 0 ||
767                strcmp(arg, "_CC=", 5) == 0)
768             /* EMPTY */;
769         else if (strcmp(arg, "_gcc=", 6) == 0 ||
770                strcmp(arg, "_g++=", 6) == 0)
771             newae(ctx->i_ae, arg + 6);
772         else
773             error(arg);
774     case '#':
775         if (arglen == 1) {
776             newae(ctx->i_ae, "-v");
777             break;
778         }
779         error(arg);
780         break;
781     case 'g':
782         newae(ctx->i_ae, "-gdwarf-2");
783         break;
784     case 'E':
785         if (arglen == 1) {
786             newae(ctx->i_ae, "-xc");
787             newae(ctx->i_ae, arg);
788             op = CW_O_PREPROCESS;
789             nolibc = 1;
790             break;
791         }
792         error(arg);
793         break;
794     case 'c':
795     case 'S':
796         if (arglen == 1) {
797             op = CW_O_COMPILE;
798             nolibc = 1;
799         }
800         /* FALLTHROUGH */
801     case 'C':
802     case 'H':
803     case 'p':
804         if (arglen == 1) {
805             newae(ctx->i_ae, arg);
806             break;
807         }
808         error(arg);
809         break;
810     case 'A':
811     case 'h':
812     case 'I':
813     case 'i':
814     case 'L':
815     case 'l':
816     case 'R':
817     case 'U':
818     case 'u':

```

```

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

```

```

876         break;
877     case 'f':
878         if (strcmp(arg, "-flags") == 0) {
879             newae(ctx->i_ae, "--help");
880             break;
881         }
882         if (strncmp(arg, "-features=zla", 13) == 0) {
883             /*
884              * Accept but ignore this -- gcc allows
885              * zero length arrays.
886              */
887             break;
888         }
889         error(arg);
890         break;
891     case 'G':
892         newae(ctx->i_ae, "--shared");
893         nolibc = 1;
894         break;
895     case 'k':
896         if (strcmp(arg, "-keeptmp") == 0) {
897             newae(ctx->i_ae, "-save-temps");
898             break;
899         }
900         error(arg);
901         break;
902     case 'K':
903         if (arglen == 1) {
904             if ((arg = *++ctx->i_oldargv) == NULL ||
905                 *arg == '\0')
906                 error("-K");
907             ctx->i_oldargc--;
908         } else {
909             arg += 2;
910         }
911         if (strcmp(arg, "pic") == 0) {
912             newae(ctx->i_ae, "-fpic");
913             pic = 1;
914             break;
915         }
916         if (strcmp(arg, "PIC") == 0) {
917             newae(ctx->i_ae, "-fPIC");
918             pic = 1;
919             break;
920         }
921         error("-K");
922         break;
923     case 'm':
924         if (strcmp(arg, "-mt") == 0) {
925             newae(ctx->i_ae, "-D_REENTRANT");
926             break;
927         }
928         if (strcmp(arg, "-m64") == 0) {
929             newae(ctx->i_ae, "-m64");
930             #if defined(__x86)
931             newae(ctx->i_ae, "-mtune=opteron");
932             #endif
933             mflag |= M64;
934             break;
935         }
936         if (strcmp(arg, "-m32") == 0) {
937             newae(ctx->i_ae, "-m32");
938             mflag |= M32;
939             break;
940         }
941         error(arg);

```

```

942         break;
943     case 'B': /* linker options */
944     case 'M':
945     case 'z':
946         {
947             char *opt;
948             size_t len;
949             char *s;
950
951             if (arglen == 1) {
952                 opt = *++ctx->i_oldargv;
953                 if (opt == NULL || *opt == '\0')
954                     error(arg);
955                 ctx->i_oldargc--;
956             } else {
957                 opt = arg + 2;
958             }
959             len = strlen(opt) + 7;
960             if ((s = malloc(len)) == NULL)
961                 nomem();
962             (void) snprintf(s, len, "-Wl,-%c%s", c, opt);
963             newae(ctx->i_ae, s);
964             free(s);
965         }
966         break;
967     case 'n':
968         if (strcmp(arg, "-noqueue") == 0) {
969             /*
970              * Horrid license server stuff - n/a
971              */
972             break;
973         }
974         error(arg);
975         break;
976     case 'O':
977         if (arglen == 1) {
978             newae(ctx->i_ae, "-O");
979             break;
980         }
981         error(arg);
982         break;
983     case 'P':
984         /*
985          * We could do '-E -o filename.i', but that's hard,
986          * and we don't need it for the case that's triggering
987          * this addition. We'll require the user to specify
988          * -o in the Makefile. If they don't they'll find out
989          * in a hurry.
990          */
991         newae(ctx->i_ae, "-E");
992         op = CW_O_PREPROCESS;
993         nolibc = 1;
994         break;
995     case 'q':
996         if (strcmp(arg, "-qp") == 0) {
997             newae(ctx->i_ae, "-p");
998             break;
999         }
1000         error(arg);
1001         break;
1002     case 's':
1003         if (arglen == 1) {
1004             newae(ctx->i_ae, "-Wl,-s");
1005             break;
1006         }
1007         error(arg);

```

```

1008         break;
1009     case 't':
1010         if (arglen == 1) {
1011             newae(ctx->i_ae, "-Wl,-t");
1012             break;
1013         }
1014         error(arg);
1015         break;
1016     case 'V':
1017         if (arglen == 1) {
1018             ctx->i_flags &= ~CW_F_ECHO;
1019             newae(ctx->i_ae, "--version");
1020             break;
1021         }
1022         error(arg);
1023         break;
1024     case 'v':
1025         if (arglen == 1) {
1026             warnings(ctx->i_ae);
1027             break;
1028         }
1029         error(arg);
1030         break;
1031     case 'W':
1032         if (strncmp(arg, "-Wp,-xc99", 9) == 0) {
1033             /*
1034              * gcc's preprocessor will accept c99
1035              * regardless, so accept and ignore.
1036              */
1037             break;
1038         }
1039         if (strncmp(arg, "-Wa,", 4) == 0 ||
1040             strncmp(arg, "-Wp,", 4) == 0 ||
1041             strncmp(arg, "-Wl,", 4) == 0) {
1042             newae(ctx->i_ae, arg);
1043             break;
1044         }
1045         if (strcmp(arg, "-W0,-xc99=pragma") == 0) {
1046             /* (undocumented) enables _Pragma */
1047             break;
1048         }
1049         if (strcmp(arg, "-W0,-xc99=%none") == 0) {
1050             /*
1051              * This is a polite way of saying
1052              * "no c99 constructs allowed!"
1053              * For now, just accept and ignore this.
1054              */
1055             break;
1056         }
1057         if (strcmp(arg, "-W0,-noglobal") == 0 ||
1058             strcmp(arg, "-W0,-xglobalstatic") == 0) {
1059             /*
1060              * gcc doesn't prefix local symbols
1061              * in debug mode, so this is not needed.
1062              */
1063             break;
1064         }
1065         if (strcmp(arg, "-W0,-Lt") == 0) {
1066             /*
1067              * Generate tests at the top of loops.
1068              * There is no direct gcc equivalent, ignore.
1069              */
1070             break;
1071         }
1072         if (strcmp(arg, "-W0,-xdbggen=no%usedonly") == 0) {
1073             newae(ctx->i_ae,

```

```

1074             "-fno-eliminate-unused-debug-symbols");
1075             newae(ctx->i_ae,
1076                 "-fno-eliminate-unused-debug-types");
1077             break;
1078         }
1079         if (strcmp(arg, "-W2,-xwrap_int") == 0) {
1080             /*
1081              * Use the legacy behaviour (pre-SS11)
1082              * for integer wrapping.
1083              * gcc does not need this.
1084              */
1085             break;
1086         }
1087         if (strcmp(arg, "-W2,-Rcond_elim") == 0) {
1088             /*
1089              * Elimination and expansion of conditionals;
1090              * gcc has no direct equivalent.
1091              */
1092             break;
1093         }
1094         if (strcmp(arg, "-Wd,-xsafe=unboundsym") == 0) {
1095             /*
1096              * Prevents optimizing away checks for
1097              * unbound weak symbol addresses. gcc does
1098              * not do this, so it's not needed.
1099              */
1100             break;
1101         }
1102         if (strncmp(arg, "-Wc,-xcode=", 11) == 0) {
1103             xlate(ctx->i_ae, arg + 11, xcode_tbl);
1104             if (strncmp(arg + 11, "pic", 3) == 0)
1105                 pic = 1;
1106             break;
1107         }
1108         if (strncmp(arg, "-Wc,-Qiselect", 13) == 0) {
1109             /*
1110              * Prevents insertion of register symbols.
1111              * gcc doesn't do this, so ignore it.
1112              */
1113             break;
1114         }
1115         if (strcmp(arg, "-Wc,-Qassembler-ounrefsym=0") == 0) {
1116             /*
1117              * Prevents optimizing away of static variables.
1118              * gcc does not do this, so it's not needed.
1119              */
1120             break;
1121         }
1122         #if defined(__x86)
1123         if (strcmp(arg, "-Wu,-xmodel=kernel") == 0) {
1124             newae(ctx->i_ae, "-ffreestanding");
1125             newae(ctx->i_ae, "-mno-red-zone");
1126             model = "-mcmmodel=kernel";
1127             nolibc = 1;
1128             break;
1129         }
1130         if (strcmp(arg, "-Wu,-save_args") == 0) {
1131             newae(ctx->i_ae, "-msave_args");
1132             break;
1133         }
1134         #endif /* __x86 */
1135         error(arg);
1136         break;
1137     case 'X':
1138         if (strcmp(arg, "-Xa") == 0 ||
1139             strcmp(arg, "-Xt") == 0) {

```



```

1140         Xamode(ctx->i_ae);
1141         break;
1142     }
1143     if (strcmp(arg, "-Xc") == 0) {
1144         Xcmode(ctx->i_ae);
1145         break;
1146     }
1147     if (strcmp(arg, "-Xs") == 0) {
1148         Xsmode(ctx->i_ae);
1149         break;
1150     }
1151     error(arg);
1152     break;
1153     case 'x':
1154         if (arglen == 1)
1155             error(arg);
1156         switch (arg[2]) {
1157 #if defined(__x86)
1158         case '3':
1159             if (strcmp(arg, "-x386") == 0) {
1160                 newae(ctx->i_ae, "-march=i386");
1161                 break;
1162             }
1163             error(arg);
1164             break;
1165         case '4':
1166             if (strcmp(arg, "-x486") == 0) {
1167                 newae(ctx->i_ae, "-march=i486");
1168                 break;
1169             }
1170             error(arg);
1171             break;
1172 #endif /* __x86 */
1173         case 'a':
1174             if (strncmp(arg, "-xarch=", 7) == 0) {
1175                 mflag |= xlate_xtb(ctx->i_ae, arg + 7);
1176                 break;
1177             }
1178             error(arg);
1179             break;
1180         case 'b':
1181             if (strncmp(arg, "-xbuiltin=", 10) == 0) {
1182                 if (strcmp(arg + 10, "%all"))
1183                     newae(ctx->i_ae, "-fbuiltin");
1184                 break;
1185             }
1186             error(arg);
1187             break;
1188         case 'C':
1189             /* Accept C++ style comments -- ignore */
1190             if (strcmp(arg, "-xCC") == 0)
1191                 break;
1192             error(arg);
1193             break;
1194         case 'c':
1195             if (strncmp(arg, "-xc99=%all", 10) == 0) {
1196                 newae(ctx->i_ae, "-std=gnu99");
1197                 break;
1198             }
1199             if (strncmp(arg, "-xc99=%none", 11) == 0) {
1200                 newae(ctx->i_ae, "-std=gnu89");
1201                 break;
1202             }
1203             if (strcmp(arg, "-xchip=", 7) == 0) {
1204                 xlate(ctx->i_ae, arg + 7, xchip_tbl);
1205                 break;

```

```

1206     }
1207     if (strncmp(arg, "-xcode=", 7) == 0) {
1208         xlate(ctx->i_ae, arg + 7, xcode_tbl);
1209         if (strncmp(arg + 7, "pic", 3) == 0)
1210             pic = 1;
1211         break;
1212     }
1213     if (strncmp(arg, "-xcache=", 8) == 0)
1214         break;
1215     if (strncmp(arg, "-xcrossfile", 11) == 0)
1216         break;
1217     error(arg);
1218     break;
1219     case 'd':
1220         if (strcmp(arg, "-xdepend") == 0)
1221             break;
1222         if (strncmp(arg, "-xdebugformat=", 14) == 0)
1223             break;
1224         error(arg);
1225         break;
1226     case 'F':
1227         /*
1228          * Compile for mapfile reordering, or unused
1229          * section elimination, syntax can be -xF or
1230          * more complex, like -xF=%all -- ignore.
1231          */
1232         if (strncmp(arg, "-xF", 3) == 0)
1233             break;
1234         error(arg);
1235         break;
1236     case 'i':
1237         if (strncmp(arg, "-xinline", 8) == 0)
1238             /* No inlining; ignore */
1239             break;
1240         if (strcmp(arg, "-xildon") == 0 ||
1241             strcmp(arg, "-xildoff") == 0)
1242             /* No incremental linking; ignore */
1243             break;
1244         error(arg);
1245         break;
1246 #if defined(__x86)
1247     case 'm':
1248         if (strcmp(arg, "-xmodel=kernel") == 0) {
1249             newae(ctx->i_ae, "-ffreestanding");
1250             newae(ctx->i_ae, "-mno-red-zone");
1251             model = "-mcmmodel=kernel";
1252             nolibc = 1;
1253             break;
1254         }
1255         error(arg);
1256         break;
1257 #endif /* __x86 */
1258     case 'M':
1259         if (strcmp(arg, "-xM") == 0) {
1260             newae(ctx->i_ae, "-M");
1261             break;
1262         }
1263         if (strcmp(arg, "-xM1") == 0) {
1264             newae(ctx->i_ae, "-MM");
1265             break;
1266         }
1267         error(arg);
1268         break;
1269     case 'n':
1270         if (strcmp(arg, "-xnolib") == 0) {
1271             nolibc = 1;

```

```

1272         break;
1273     }
1274     error(arg);
1275     break;
1276 case 'O':
1277     if (strncmp(arg, "-xO", 3) == 0) {
1278         size_t len = strlen(arg);
1279         char *s = NULL;
1280         int c = *(arg + 3);
1281         int level;
1282
1283         if (len != 4 || !isdigit(c))
1284             error(arg);
1285
1286         level = atoi(arg + 3);
1287         if (level > 5)
1288             error(arg);
1289         if (level >= 2) {
1290             /*
1291              * For gcc-3.4.x at -O2 we
1292              * need to disable optimizations
1293              * that break ON.
1294              */
1295             optim_disable(ctx->i_ae, level);
1296             /*
1297              * limit -xO3 to -O2 as well.
1298              */
1299             level = 2;
1300         }
1301         if (asprintf(&s, "-O%d", level) == -1)
1302             nomem();
1303         newae(ctx->i_ae, s);
1304         free(s);
1305         break;
1306     }
1307     error(arg);
1308     break;
1309 case 'p':
1310     if (strcmp(arg, "-xpentium") == 0) {
1311         newae(ctx->i_ae, "-march=pentium");
1312         break;
1313     }
1314     if (strcmp(arg, "-xpg") == 0) {
1315         newae(ctx->i_ae, "-pg");
1316         break;
1317     }
1318     error(arg);
1319     break;
1320 case 'r':
1321     if (strncmp(arg, "-xregs=", 7) == 0) {
1322         xlate(ctx->i_ae, arg + 7, xregs_tbl);
1323         break;
1324     }
1325     error(arg);
1326     break;
1327 case 's':
1328     if (strcmp(arg, "-xs") == 0 ||
1329         strcmp(arg, "-xspace") == 0 ||
1330         strcmp(arg, "-xstrconst") == 0)
1331         break;
1332     error(arg);
1333     break;
1334 case 't':
1335     if (strcmp(arg, "-xtransition") == 0) {
1336         newae(ctx->i_ae, "-Wtransition");
1337         break;

```

```

1338     }
1339     if (strcmp(arg, "-xtrigraphs=yes") == 0) {
1340         newae(ctx->i_ae, "-trigraphs");
1341         break;
1342     }
1343     if (strcmp(arg, "-xtrigraphs=no") == 0) {
1344         newae(ctx->i_ae, "-notrigraphs");
1345         break;
1346     }
1347     if (strncmp(arg, "-xtarget=", 9) == 0) {
1348         xlate(ctx->i_ae, arg + 9, xtarget_tbl);
1349         break;
1350     }
1351     error(arg);
1352     break;
1353 case 'e':
1354 case 'h':
1355 case 'l':
1356 default:
1357     error(arg);
1358     break;
1359 }
1360 break;
1361 case 'Y':
1362     if (arglen == 1) {
1363         if ((arg = **++ctx->i_oldargv) == NULL ||
1364             *arg == '\\0')
1365             error("-Y");
1366         ctx->i_oldargc--;
1367         arglen = strlen(arg + 1);
1368     } else {
1369         arg += 2;
1370     }
1371     /* Just ignore -YS,... for now */
1372     if (strncmp(arg, "S", 2) == 0)
1373         break;
1374     if (strncmp(arg, "l", 2) == 0) {
1375         char *s = strdup(arg);
1376         s[0] = '-';
1377         s[1] = 'B';
1378         newae(ctx->i_ae, s);
1379         free(s);
1380         break;
1381     }
1382     if (strncmp(arg, "I", 2) == 0) {
1383         char *s = strdup(arg);
1384         s[0] = '-';
1385         s[1] = 'I';
1386         newae(ctx->i_ae, "-nostdinc");
1387         newae(ctx->i_ae, s);
1388         free(s);
1389         break;
1390     }
1391     error(arg);
1392     break;
1393 case 'Q':
1394     /*
1395      * We could map -Qy into -Wl,-Qy etc.
1396      */
1397 default:
1398     error(arg);
1399     break;
1400 }
1401 }
1402 }
1403     free(nameflag);

```

```

1405     if (c_files > 1 && (ctx->i_flags & CW_F_SHADOW) &&
1406         op != CW_O_PREPROCESS) {
1407         errx(2, "multiple source files are "
1408             "allowed only with -E or -P");
1426         (void) fprintf(stderr, "%s: error: multiple source files are "
1427             "allowed only with -E or -P\n", progname);
1428         exit(2);
1409     }

1411     /*
1412     * Make sure that we do not have any unintended interactions between
1413     * the xarch options passed in and the version of the Studio compiler
1414     * used.
1415     */
1416     if ((mflag & (SS11|SS12)) == (SS11|SS12)) {
1417         errx(2,
1437             (void) fprintf(stderr,
1418                 "Conflicting \"-xarch=\" flags (both Studio 11 and 12)\n");
1439             exit(2);
1419     }

1421     switch (mflag) {
1422     case 0:
1423         /* FALLTHROUGH */
1424         case M32:
1425     #if defined(__sparc)
1426         /*
1427         * Only -m32 is defined and so put in the missing xarch
1428         * translation.
1429         */
1430         newae(ctx->i_ae, "-mcpu=v8");
1431         newae(ctx->i_ae, "-mno-v8plus");
1432     #endif
1433         break;
1434     case M64:
1435     #if defined(__sparc)
1436         /*
1437         * Only -m64 is defined and so put in the missing xarch
1438         * translation.
1439         */
1440         newae(ctx->i_ae, "-mcpu=v9");
1441     #endif
1442         break;
1443     case SS12:
1444     #if defined(__sparc)
1445         /* no -m32/-m64 flag used - this is an error for sparc builds */
1446         (void) fprintf(stderr, "No -m32/-m64 flag defined\n");
1447         exit(2);
1448     #endif
1449         break;
1450     case SS11:
1451         /* FALLTHROUGH */
1452     case (SS11|M32):
1453     case (SS11|M64):
1454         break;
1455     case (SS12|M32):
1456     #if defined(__sparc)
1457         /*
1458         * Need to add in further 32 bit options because with SS12
1459         * the xarch=sparcvis option can be applied to 32 or 64
1460         * bit, and so the translation table (xtbl) cannot handle
1461         * that.
1462         */
1463         newae(ctx->i_ae, "-mv8plus");
1464     #endif

```

```

1465         break;
1466     case (SS12|M64):
1467         break;
1468     default:
1469         (void) fprintf(stderr,
1470             "Incompatible -xarch= and/or -m32/-m64 options used.\n");
1471         exit(2);
1472     }

1474     if ((op == CW_O_LINK || op == CW_O_PREPROCESS) &&
1475         (ctx->i_flags & CW_F_SHADOW))
1494     if (op == CW_O_LINK && (ctx->i_flags & CW_F_SHADOW))
1476         exit(0);

1478     if (model && !pic)
1479         newae(ctx->i_ae, model);
1480     if (!nolibc)
1481         newae(ctx->i_ae, "-lc");
1482     if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1483         newae(ctx->i_ae, "-o");
1484         newae(ctx->i_ae, ctx->i_discard);
1485     }
1486 }

1488 static void
1489 do_cc(cw_ictx_t *ctx)
1490 {
1491     int in_output = 0, seen_o = 0;
1492     cw_op_t op = CW_O_LINK;
1493     char *nameflag;

1495     if (ctx->i_flags & CW_F_PROG) {
1496         newae(ctx->i_ae, "-V");
1497         return;
1498     }

1500     if (asprintf(&nameflag, "-_s=", ctx->i_compiler->c_name) == -1)
1501         nomem();

1503     while (--ctx->i_oldargc > 0) {
1504         char *arg = ++ctx->i_oldargv;

1506         if (strncmp(arg, "-_cc=", 5) == 0) {
1507             newae(ctx->i_ae, strchr(arg, '=') + 1);
1508             continue;
1509         }

1511         if (*arg != '-') {
1512             if (in_output == 0 || !(ctx->i_flags & CW_F_SHADOW)) {
1513                 newae(ctx->i_ae, arg);
1514             } else {
1515                 in_output = 0;
1516                 newae(ctx->i_ae, ctx->i_discard);
1517             }
1518             continue;
1519         }
1520         switch (*(arg + 1)) {
1521         case '_':
1522             if ((strncmp(arg, nameflag, strlen(nameflag)) == 0) ||
1523                 (strncmp(arg, "-_cc=", 5) == 0) ||
1524                 (strncmp(arg, "-_sun=", 6) == 0)) {
1525                 newae(ctx->i_ae, strchr(arg, '=') + 1);
1526             }
1527             if (strncmp(arg, "-_noecho") == 0) {
1528                 ctx->i_flags &= ~CW_F_ECHO;
1529             }
1530             else if (strncmp(arg, "-_cc=", 5) == 0 ||
1531                 strncmp(arg, "-_cc=", 5) == 0) {

```

```

1536     newae(ctx->i_ae, arg + 5);
1537     } else if (strncmp(arg, "-_gcc=", 6) != 0 &&
1538     strncmp(arg, "-_g++=", 6) != 0) {
1539         (void) fprintf(stderr,
1540         "%s: invalid argument '%s'\n", progname,
1541         arg);
1542         exit(2);
1543     }
1544     break;
1545
1546 case 'V':
1547     ctx->i_flags &= ~CW_F_ECHO;
1548     newae(ctx->i_ae, arg);
1549     break;
1550 case 'o':
1551     seen_o = 1;
1552     if (strlen(arg) == 2) {
1553         in_output = 1;
1554         newae(ctx->i_ae, arg);
1555     } else if (ctx->i_flags & CW_F_SHADOW) {
1556         newae(ctx->i_ae, "-o");
1557     } else {
1558         newae(ctx->i_ae, ctx->i_discard);
1559     }
1560     break;
1561 case 'c':
1562 case 'S':
1563     if (strlen(arg) == 2)
1564         op = CW_O_COMPILE;
1565     newae(ctx->i_ae, arg);
1566     break;
1567 case 'E':
1568 case 'P':
1569     if (strlen(arg) == 2)
1570         op = CW_O_PREPROCESS;
1571     /*FALLTHROUGH*/
1572     default:
1573         newae(ctx->i_ae, arg);
1574 }
1575
1576 free(nameflag);
1577
1578 if ((op == CW_O_LINK || op == CW_O_PREPROCESS) &&
1579 (ctx->i_flags & CW_F_SHADOW))
1580     exit(0);
1581
1582 if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1583     newae(ctx->i_ae, "-o");
1584     newae(ctx->i_ae, ctx->i_discard);
1585 }
1586
1587 static void
1588 prepctx(cw_ictx_t *ctx)
1589 {
1590     newae(ctx->i_ae, ctx->i_compiler->c_path);
1591     const char *dir = NULL, *cmd;
1592     char *program = NULL;
1593     size_t len;
1594
1595     switch (CIDX(CC(ctx), ctx->i_flags)) {
1596     case CIDX(CW_C_CC, 0):
1597         program = getenv("CW_CC");
1598         dir = getenv("CW_CC_DIR");

```

```

1599         break;
1600     case CIDX(CW_C_CC, CW_F_CXX):
1601         program = getenv("CW_CPLUSPLUS");
1602         dir = getenv("CW_CPLUSPLUS_DIR");
1603         break;
1604     case CIDX(CW_C_GCC, 0):
1605         program = getenv("CW_GCC");
1606         dir = getenv("CW_GCC_DIR");
1607         break;
1608     case CIDX(CW_C_GCC, CW_F_CXX):
1609         program = getenv("CW_GPLUSPLUS");
1610         dir = getenv("CW_GPLUSPLUS_DIR");
1611         break;
1612     }
1613
1614     if (program == NULL) {
1615         if (dir == NULL)
1616             dir = default_dir[CC(ctx)];
1617         cmd = cmds[CIDX(CC(ctx), ctx->i_flags)];
1618         len = strlen(dir) + strlen(cmd) + 2;
1619         if ((program = malloc(len)) == NULL)
1620             nomem();
1621         (void) snprintf(program, len, "%s/%s", dir, cmd);
1622     }
1623     newae(ctx->i_ae, program);
1624
1625     if (ctx->i_flags & CW_F_PROG) {
1626         (void) printf("%s: %s\n", (ctx->i_flags & CW_F_SHADOW) ?
1627         "shadow" : "primary", ctx->i_compiler->c_path);
1628         "shadow" : "primary", program);
1629         (void) fflush(stdout);
1630     }
1631
1632     if (!(ctx->i_flags & CW_F_XLATE))
1633         return;
1634
1635     switch (ctx->i_compiler->c_style) {
1636     case SUN:
1637         switch (CC(ctx)) {
1638         case CW_C_CC:
1639             do_cc(ctx);
1640             break;
1641         case GNU:
1642         case CW_C_GCC:
1643             do_gcc(ctx);
1644             break;
1645         }
1646     }
1647
1648     static int
1649     invoke(cw_ictx_t *ctx)
1650     {
1651         char **newargv;
1652         int ac;
1653         struct ae *a;
1654
1655         if ((newargv = calloc(sizeof(*newargv), ctx->i_ae->ael_argc + 1)) ==
1656         NULL)
1657             nomem();
1658
1659         if (ctx->i_flags & CW_F_ECHO)
1660             (void) fprintf(stderr, "+ ");
1661
1662         for (ac = 0, a = ctx->i_ae->ael_head; a; a = a->ae_next, ac++) {
1663             newargv[ac] = a->ae_arg;

```

```

1613         if (ctx->i_flags & CW_F_ECHO)
1614             (void) fprintf(stderr, "%s ", a->ae_arg);
1615         if (a == ctx->i_ae->ael_tail)
1616             break;
1617     }
1618
1619     if (ctx->i_flags & CW_F_ECHO) {
1620         (void) fprintf(stderr, "\n");
1621         (void) fflush(stderr);
1622     }
1623
1624     if (!(ctx->i_flags & CW_F_EXEC))
1625         return (0);
1626
1627     /*
1628     * We must fix up the environment here so that the dependency files are
1629     * not trampled by the shadow compiler. Also take care of GCC
1630     * environment variables that will throw off gcc. This assumes a primary
1631     * gcc.
1632     * We must fix up the environment here so that the
1633     * dependency files are not trampled by the shadow compiler.
1634     */
1635     if ((ctx->i_flags & CW_F_SHADOW) &&
1636         (unsetenv("SUNPRO_DEPENDENCIES") != 0 ||
1637          unsetenv("DEPENDENCIES_OUTPUT") != 0 ||
1638          unsetenv("GCC_ROOT") != 0) {
1639         unsetenv("DEPENDENCIES_OUTPUT") != 0)) {
1640             (void) fprintf(stderr, "error: environment setup failed: %s\n",
1641                          strerror(errno));
1642             return (-1);
1643         }
1644     }
1645
1646     (void) execv(newargv[0], newargv);
1647     warn("couldn't run %s", newargv[0]);
1648     cw_perror("couldn't run %s", newargv[0]);
1649
1650     return (-1);
1651 }
1652
1653 static int
1654 reap(cw_ictx_t *ctx)
1655 {
1656     int status, ret = 0;
1657     char buf[1024];
1658     struct stat s;
1659
1660     /*
1661     * Only wait for one specific child.
1662     */
1663     if (ctx->i_pid <= 0)
1664         return (-1);
1665
1666     do {
1667         if (waitpid(ctx->i_pid, &status, 0) < 0) {
1668             warn("cannot reap child");
1669             cw_perror("cannot reap child");
1670             return (-1);
1671         }
1672         if (status != 0) {
1673             if (WIFSIGNALED(status)) {
1674                 ret = -WTERMSIG(status);
1675                 break;
1676             } else if (WIFEXITED(status)) {
1677                 ret = WEXITSTATUS(status);
1678                 break;
1679             }
1680         }
1681     }

```

```

1674     } while (!WIFEXITED(status) && !WIFSIGNALED(status));
1675
1676     (void) unlink(ctx->i_discard);
1677
1678     if (stat(ctx->i_stderr, &s) < 0) {
1679         warn("stat failed on child cleanup");
1680         cw_perror("stat failed on child cleanup");
1681         return (-1);
1682     }
1683     if (s.st_size != 0) {
1684         FILE *f;
1685
1686         if ((f = fopen(ctx->i_stderr, "r")) != NULL) {
1687             while (fgets(buf, sizeof (buf), f))
1688                 (void) fprintf(stderr, "%s", buf);
1689             (void) fflush(stderr);
1690             (void) fclose(f);
1691         }
1692     }
1693     (void) unlink(ctx->i_stderr);
1694     free(ctx->i_stderr);
1695
1696     /*
1697     * cc returns an error code when given -V; we want that to succeed.
1698     */
1699     if (ctx->i_flags & CW_F_PROG)
1700         return (0);
1701
1702     return (ret);
1703 }
1704
1705 static int
1706 exec_ctx(cw_ictx_t *ctx, int block)
1707 {
1708     char *file;
1709
1710     /*
1711     * To avoid offending cc's sensibilities, the name of its output
1712     * file must end in '.o'.
1713     */
1714     if ((file = tempnam(NULL, ".cw")) == NULL) {
1715         nomem();
1716         return (-1);
1717     }
1718     (void) strcpy(ctx->i_discard, file, MAXPATHLEN);
1719     (void) strcat(ctx->i_discard, ".o", MAXPATHLEN);
1720     free(file);
1721
1722     if ((ctx->i_stderr = tempnam(NULL, ".cw")) == NULL) {
1723         nomem();
1724         return (-1);
1725     }
1726
1727     if ((ctx->i_pid = fork()) == 0) {
1728         int fd;
1729
1730         (void) fclose(stderr);
1731         if ((fd = open(ctx->i_stderr, O_WRONLY | O_CREAT | O_EXCL,
1732                      0666)) < 0) {
1733             err(1, "open failed for standard error");
1734             cw_perror("open failed for standard error");
1735             exit(1);
1736         }
1737         if (dup2(fd, 2) < 0) {
1738             err(1, "dup2 failed for standard error");

```

```

1781         cw_perror("dup2 failed for standard error");
1782         exit(1);
1737     }
1738     if (fd != 2)
1739         (void) close(fd);
1740     if (freopen("/dev/fd/2", "w", stderr) == NULL) {
1741         err(1, "freopen failed for /dev/fd/2");
1787         cw_perror("freopen failed for /dev/fd/2");
1788         exit(1);
1742     }

1744     prepctx(ctx);
1745     exit(invoke(ctx));
1746 }

1748     if (ctx->i_pid < 0) {
1749         err(1, "fork failed");
1795         cw_perror("fork failed");
1796         return (1);
1750     }

1752     if (block)
1753         return (reap(ctx));

1755     return (0);
1756 }

1758 static void
1759 parse_compiler(const char *spec, cw_compiler_t *compiler)
1760 {
1761     char *tspec, *token;

1763     if ((tspec = strdup(spec)) == NULL)
1764         nomem();

1766     if ((token = strsep(&tspec, ",")) == NULL)
1767         errx(1, "Compiler is missing a name: %s", spec);
1768     compiler->c_name = token;

1770     if ((token = strsep(&tspec, ",")) == NULL)
1771         errx(1, "Compiler is missing a path: %s", spec);
1772     compiler->c_path = token;

1774     if ((token = strsep(&tspec, ",")) == NULL)
1775         errx(1, "Compiler is missing a style: %s", spec);

1777     if ((strcasecmp(token, "gnu") == 0) ||
1778         (strcasecmp(token, "gcc") == 0))
1779         compiler->c_style = GNU;
1780     else if ((strcasecmp(token, "sun") == 0) ||
1781             (strcasecmp(token, "cc") == 0))
1782         compiler->c_style = SUN;
1783     else
1784         errx(1, "unknown compiler style: %s", token);

1786     if (tspec != NULL)
1787         errx(1, "Excess tokens in compiler: %s", spec);
1788 }

1790 int
1791 main(int argc, char **argv)
1792 {
1793     int ch;
1794     cw_compiler_t primary = { NULL, NULL, 0 };
1795     cw_compiler_t shadows[10];
1796     int nshadows = 0;

```

```

1808     cw_ictx_t *ctx = newictx();
1809     cw_ictx_t *ctx_shadow = newictx();
1810     const char *dir;
1811     int do_serial, do_shadow;
1797     int ret = 0;
1798     boolean_t do_serial = B_FALSE;
1799     boolean_t do_exec = B_FALSE;
1800     boolean_t vflg = B_FALSE;
1801     boolean_t Cflg = B_FALSE;
1802     boolean_t cflg = B_FALSE;
1803     boolean_t nflg = B_FALSE;

1805     cw_ictx_t *main_ctx;
1814     if ((progname = strrchr(argv[0], '/')) == NULL)
1815         progname = argv[0];
1816     else
1817         progname++;

1807     static struct option longopts[] = {
1808         { "compiler", no_argument, NULL, 'c' },
1809         { "noecho", no_argument, NULL, 'n' },
1810         { "primary", required_argument, NULL, 'p' },
1811         { "shadow", required_argument, NULL, 's' },
1812         { "versions", no_argument, NULL, 'v' },
1813         { NULL, 0, NULL, 0 },
1814     };

1817     if ((main_ctx = newictx()) == NULL)
1819         if (ctx == NULL || ctx_shadow == NULL)
1818             nomem();

1820     while ((ch = getopt_long(argc, argv, "C", longopts, NULL)) != -1) {
1821         switch (ch) {
1822             case 'c':
1823                 cflg = B_TRUE;
1824                 break;
1825             case 'C':
1826                 Cflg = B_TRUE;
1827                 break;
1828             case 'n':
1829                 nflg = B_TRUE;
1830                 break;
1831             case 'p':
1832                 if (primary.c_path != NULL) {
1833                     warnx("Only one primary compiler may "
1834                          "be specified");
1835                     usage();
1836                 }
1822             ctx->i_flags = CW_F_ECHO|CW_F_XLATE;

1838         parse_compiler(optarg, &primary);
1839         break;
1840     case 's':
1841         if (nshadows >= 10)
1842             errx(1, "May only use 10 shadows at "
1843                  "the moment");
1844         parse_compiler(optarg, &shadows[nshadows]);
1845         nshadows++;
1846         break;
1847     case 'v':
1848         vflg = B_TRUE;
1849         break;
1850     default:
1851         (void) fprintf(stderr, "Did you forget '--'? \n");
1852         usage();

```

```

1824 /*
1825  * Figure out where to get our tools from. This depends on
1826  * the environment variables set at run time.
1827  */
1828 if ((dir = getenv("SPRO_VROOT")) != NULL) {
1829     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1830                    "%s/bin", dir);
1831 } else if ((dir = getenv("SPRO_ROOT")) != NULL) {
1832     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1833                    "%s/SS12/bin", dir);
1834 } else if ((dir = getenv("BUILD_TOOLS")) != NULL) {
1835     (void) snprintf(default_dir[CW_C_CC], MAXPATHLEN,
1836                    "%s/SUNWspro/SS12/bin", dir);
1837 }
1838 }
1839
1840 if (primary.c_path == NULL) {
1841     warnx("A primary compiler must be specified");
1842     usage();
1843 }
1844 if ((dir = getenv("GNUC_ROOT")) != NULL) {
1845     (void) snprintf(default_dir[CW_C_GCC], MAXPATHLEN,
1846                    "%s/bin", dir);
1847 }
1848
1849 do_serial = (getenv("CW_SHADOW_SERIAL") == NULL) ? B_FALSE : B_TRUE;
1850 do_exec = (getenv("CW_NO_EXEC") == NULL) ? B_TRUE : B_FALSE;
1851 do_shadow = (getenv("CW_NO_SHADOW") ? 0 : 1);
1852 do_serial = (getenv("CW_SHADOW_SERIAL") ? 1 : 0);
1853
1854 /* Leave room for argv[0] */
1855 argc -= (optind - 1);
1856 argv += (optind - 1);
1857 if (getenv("CW_NO_EXEC") == NULL)
1858     ctx->i_flags |= CW_F_EXEC;
1859
1860 main_ctx->i_oldargc = argc;
1861 main_ctx->i_oldargv = argv;
1862 main_ctx->i_flags = CW_F_XLATE;
1863 if (nflg == 0)
1864     main_ctx->i_flags |= CW_F_ECHO;
1865 if (do_exec)
1866     main_ctx->i_flags |= CW_F_EXEC;
1867 if (Cflg)
1868     main_ctx->i_flags |= CW_F_CXX;
1869 main_ctx->i_compiler = &primary;
1870 /*
1871  * The first argument must be one of "--cc", "--gcc", "--CC", or "--g++"
1872  */
1873 if (argc == 1)
1874     usage();
1875 argc--;
1876 argv++;
1877 if (strcmp(argv[0], "--cc") == 0) {
1878     ctx->i_compiler = CW_C_CC;
1879 } else if (strcmp(argv[0], "--gcc") == 0) {
1880     ctx->i_compiler = CW_C_GCC;
1881 } else if (strcmp(argv[0], "--CC") == 0) {
1882     ctx->i_compiler = CW_C_CC;
1883     ctx->i_flags |= CW_F_CXX;
1884 } else if (strcmp(argv[0], "--g++") == 0) {
1885     ctx->i_compiler = CW_C_GCC;
1886     ctx->i_flags |= CW_F_CXX;
1887 } else {
1888     /* assume "--gcc" by default */
1889     argc++;
1890     argv--;

```

```

1871     ctx->i_compiler = CW_C_GCC;
1872 }
1873
1874 if (cflg) {
1875     (void) fputs(primary.c_path, stdout);
1876     /*
1877      * --compiler - tell us the path to the primary compiler only
1878      */
1879     if (argc > 1 && strcmp(argv[1], "--compiler") == 0) {
1880         ctx->i_flags &= ~CW_F_XLATE;
1881         prepctx(ctx);
1882         (void) printf("%s\n", ctx->i_ae->ael_head->ae_arg);
1883         return (0);
1884     }
1885 }
1886
1887 if (vflg) {
1888     (void) printf("cw version %s\n", CW_VERSION);
1889     /*
1890      * --versions - tell us the cw version, paths to all compilers, and
1891      * ask each for its version if we know how.
1892      */
1893     if (argc > 1 && strcmp(argv[1], "--versions") == 0) {
1894         (void) printf("cw version %s", CW_VERSION);
1895         if (!do_shadow)
1896             (void) printf(" (SHADOW MODE DISABLED)");
1897         (void) printf("\n");
1898         (void) fflush(stdout);
1899         main_ctx->i_flags &= ~CW_F_ECHO;
1900         main_ctx->i_flags |= CW_F_PROG | CW_F_EXEC;
1901         ctx->i_flags &= ~CW_F_ECHO;
1902         ctx->i_flags |= CW_F_PROG | CW_F_EXEC;
1903         argc--;
1904         argv++;
1905         do_serial = 1;
1906     }
1907 }
1908
1909 ret |= exec_ctx(main_ctx, do_serial);
1910 ctx->i_oldargc = argc;
1911 ctx->i_oldargv = argv;
1912
1913 for (int i = 0; i < nshadows; i++) {
1914     int r;
1915     cw_ictx_t *shadow_ctx;
1916     ret |= exec_ctx(ctx, do_serial);
1917 }
1918
1919 if ((shadow_ctx = newctx()) == NULL)
1920     nomem();
1921
1922 memcpy(shadow_ctx, main_ctx, sizeof (cw_ictx_t));
1923
1924 shadow_ctx->i_flags |= CW_F_SHADOW;
1925 shadow_ctx->i_compiler = &shadows[i];
1926
1927 r = exec_ctx(shadow_ctx, do_serial);
1928 if (r == 0) {
1929     shadow_ctx->i_next = main_ctx->i_next;
1930     main_ctx->i_next = shadow_ctx;
1931 }
1932 if (do_shadow) {
1933     (void) memcpy(ctx_shadow, ctx, sizeof (cw_ictx_t));
1934     ctx_shadow->i_flags |= CW_F_SHADOW;
1935     ret |= exec_ctx(ctx_shadow, 1);
1936 }
1937
1938 ret |= r;
1939 }
1940
1941 if (!do_serial) {

```

```
1914         cw_ictx_t *next = main_ctx;
1915         while (next != NULL) {
1916             cw_ictx_t *toreap = next;
1917             next = next->i_next;
1918             ret |= reap(toreap);
1919         }
1920     }
1921     if (!do_serial)
1912         ret |= reap(ctx);
1913
1922     return (ret);
1923 }
unchanged_portion_omitted
```


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

1

```
*****
9567 Wed Oct 3 20:38:56 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: Andy Fiddaman <omnios@citrus-it.net>
*****
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"
141 export MULTI_PROTO="no"

143 #
```

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

2

```
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_VROOT
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/scripts/nightly.sh

1

```
*****
58253 Wed Oct 3 20:38:57 2018
new/usr/src/tools/scripts/nightly.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: Andy Fiddaman <omnios@citrus-it.net>
*****
_____unchanged_portion_omitted_____

1588 #
1589 #       Decide whether to bringover to the codemgr workspace
1590 #
1591 if [ "$n_FLAG" = "n" ]; then
1592     PARENT_SCM_TYPE=$(parent_wstype)
1593
1594     if [[ $SCM_TYPE != none && $SCM_TYPE != $PARENT_SCM_TYPE ]]; then
1595         echo "cannot bringover from $PARENT_SCM_TYPE to $SCM_TYPE, " \
1596             "quitting at 'date'." | tee -a $mail_msg_file >> $LOGFILE
1597         exit 1
1598     fi
1599
1600     run_hook PRE_BRINGOVER
1601
1602     echo "\n=== bringover to $CODEMGR_WS at 'date' ===\n" >> $LOGFILE
1603     echo "\n=== BRINGOVER LOG ===\n" >> $mail_msg_file
1604
1605     eval "bringover_${PARENT_SCM_TYPE}" 2>&1 |
1606         tee -a $mail_msg_file >> $LOGFILE
1607
1608     if [ -f $TMPDIR/bringover_failed ]; then
1609         rm -f $TMPDIR/bringover_failed
1610         build_ok=n
1611         echo "trouble with bringover, quitting at 'date'." |
1612             tee -a $mail_msg_file >> $LOGFILE
1613         exit 1
1614     fi
1615
1616     #
1617     # It's possible that we used the bringover above to create
1618     # $CODEMGR_WS.  If so, then SCM_TYPE was previously "none,"
1619     # but should now be the same as $BRINGOVER_WS.
1620     #
1621     [[ $SCM_TYPE = none ]] && SCM_TYPE=$PARENT_SCM_TYPE
1622
1623     run_hook POST_BRINGOVER
1624
1625     check_closed_bins
1626
1627 else
1628     echo "\n=== No bringover to $CODEMGR_WS ===\n" >> $LOGFILE
1629 fi
1630
1631 # Safeguards
1632 [[ -v CODEMGR_WS ]] || fatal_error "Error: Variable CODEMGR_WS not set."
1633 [[ -d "${CODEMGR_WS}" ]] || fatal_error "Error: ${CODEMGR_WS} is not a directory"
1634 [[ -f "${CODEMGR_WS}/usr/src/Makefile" ]] || fatal_error "Error: ${CODEMGR_WS}/u
1635
1636 if [[ "$t_FLAG" = "y" ]]; then
1637     echo "\n=== Bootstrapping cw ===\n" >> $LOGFILE
1638     ( cd ${TOOLS}
1639         set_non_debug_build_flags
1640         rm -f $TMPDIR/make-state
1641         $MAKE -K $TMPDIR/make-state -e TARGET=install cw 2>&1 >> $LOGFILE
1642         [[ "$?" -ne 0 ]] && fatal_error "Error: could not bootstrap cw"
```

new/usr/src/tools/scripts/nightly.sh

2

```
1643     )
1644
1645     # Switch ONBLD_TOOLS early if -t is specified so that
1646     # we could use bootstrapped cw for compiler checks.
1647     ONBLD_TOOLS=${TOOLS_PROTO}/opt/onbld
1648     export ONBLD_TOOLS
1649 fi
1650
1651 echo "\n=== Build environment ===\n" | tee -a $build_environ_file >> $LOGFILE
1652
1653 # System
1654 whence uname | tee -a $build_environ_file >> $LOGFILE
1655 uname -a 2>&1 | tee -a $build_environ_file >> $LOGFILE
1656 echo | tee -a $build_environ_file >> $LOGFILE
1657
1658 # make
1659 whence $MAKE | tee -a $build_environ_file >> $LOGFILE
1660 $MAKE -v | tee -a $build_environ_file >> $LOGFILE
1661 echo "number of concurrent jobs = $DMAKE_MAX_JOBS" |
1662     tee -a $build_environ_file >> $LOGFILE
1663
1664 #
1665 # Report the compiler versions.
1666 #
1667
1668 if [[ ! -f $SRC/Makefile ]]; then
1669     build_ok=n
1670     echo "\nUnable to find \"Makefile\" in $SRC." | \
1671         tee -a $build_environ_file >> $LOGFILE
1672     exit 1
1673 fi
1674
1675 ( cd $SRC
1676     for target in cc-version java-version openssl-version; do
1677         for target in cc-version cc64-version java-version openssl-version; do
1678             echo
1679             # Put statefile somewhere we know we can write to rather than trip
1680             # over a read-only $srcroot.
1681             #
1682             rm -f $TMPDIR/make-state
1683             export SRC
1684             if $MAKE -K $TMPDIR/make-state -e $target 2>/dev/null; then
1685                 continue
1686             fi
1687             touch $TMPDIR/nocompiler
1688         done
1689         echo
1690     ) | tee -a $build_environ_file >> $LOGFILE
1691
1692 if [ -f $TMPDIR/nocompiler ]; then
1693     rm -f $TMPDIR/nocompiler
1694     build_ok=n
1695     echo "Aborting due to missing compiler." |
1696         tee -a $build_environ_file >> $LOGFILE
1697     exit 1
1698 fi
1699
1700 # as
1701 whence as | tee -a $build_environ_file >> $LOGFILE
1702 as -V 2>&1 | head -1 | tee -a $build_environ_file >> $LOGFILE
1703 echo | tee -a $build_environ_file >> $LOGFILE
1704
1705 # Check that we're running a capable link-editor
1706 whence ld | tee -a $build_environ_file >> $LOGFILE
1707 LDVER='ld -V 2>&1'
```

new/usr/src/tools/scripts/nightly.sh

3

```

1708 echo $LDVER | tee -a $build_envIRON_file >> $LOGFILE
1709 LDVER=`echo $LDVER | sed -e "s/.*-1\.\([0-9]*\)\.*/\1/"`
1710 if [ `expr $LDVER \< 422` -eq 1 ]; then
1711     echo "The link-editor needs to be at version 422 or higher to build" | \
1712     tee -a $build_envIRON_file >> $LOGFILE
1713     echo "the latest stuff. Hope your build works." | \
1714     tee -a $build_envIRON_file >> $LOGFILE
1715 fi

1717 #
1718 # Build and use the workspace's tools if requested
1719 #
1720 if [[ "$t_FLAG" = "y" ]]; then
1721     set_non_debug_build_flags

1723     build_tools ${TOOLS_PROTO}
1724     if (( $? != 0 )); then
1725         build_ok=n
1726     else
1727         STABS=${TOOLS_PROTO}/opt/onbld/bin/${MACH}/stabs
1728         export STABS
1729         CTFSTABS=${TOOLS_PROTO}/opt/onbld/bin/${MACH}/ctfstabs
1730         export CTFSTABS
1731         GENOFFSETS=${TOOLS_PROTO}/opt/onbld/bin/genoffsets
1732         export GENOFFSETS

1734         CTFCONVERT=${TOOLS_PROTO}/opt/onbld/bin/${MACH}/ctfconvert
1735         export CTFCONVERT
1736         CTFMERGE=${TOOLS_PROTO}/opt/onbld/bin/${MACH}/ctfmerge
1737         export CTFMERGE

1739         PATH=${TOOLS_PROTO}/opt/onbld/bin/${MACH}:${PATH}"
1740         PATH=${TOOLS_PROTO}/opt/onbld/bin:${PATH}"
1741         export PATH

1743         echo "\n=== New environment settings. ===\n" >> $LOGFILE
1744         echo "STABS=${STABS}" >> $LOGFILE
1745         echo "CTFSTABS=${CTFSTABS}" >> $LOGFILE
1746         echo "CTFCONVERT=${CTFCONVERT}" >> $LOGFILE
1747         echo "CTFMERGE=${CTFMERGE}" >> $LOGFILE
1748         echo "PATH=${PATH}" >> $LOGFILE
1749         echo "ONBLD_TOOLS=${ONBLD_TOOLS}" >> $LOGFILE
1750     fi
1751 fi

1753 # timestamp the start of the normal build; the findunref tool uses it.
1754 touch $SRC/.build.tstamp

1756 normal_build

1758 ORIG_SRC=$SRC
1759 BINARCHIVE=${CODEMGR_WS}/bin-${MACH}.cpio.Z

1762 #
1763 # There are several checks that need to look at the proto area, but
1764 # they only need to look at one, and they don't care whether it's
1765 # DEBUG or non-DEBUG.
1766 #
1767 if [[ "$MULTI_PROTO" = yes && "$D_FLAG" = n ]]; then
1768     checkroot=$ROOT-nd
1769 else
1770     checkroot=$ROOT
1771 fi

1773 if [ "$build_ok" = "y" ]; then

```

new/usr/src/tools/scripts/nightly.sh

4

```

1774     echo "\n=== Creating protolist system file at `date` ===" \
1775     >> $LOGFILE
1776     protolist $checkroot > $ATLOG/proto_list_${MACH}
1777     echo "=== protolist system file created at `date` ===\n" \
1778     >> $LOGFILE

1780     if [ "$N_FLAG" != "y" ]; then

1782         E1=
1783         f1=
1784         for f in $f1; do
1785             if [ -f "$f" ]; then
1786                 E1="$E1 -e $f"
1787             fi
1788         done

1790         E2=
1791         f2=
1792         if [ -d "$SRC/pkg" ]; then
1793             f2="$f2 exceptions/packaging"
1794         fi

1796         for f in $f2; do
1797             if [ -f "$f" ]; then
1798                 E2="$E2 -e $f"
1799             fi
1800         done
1801     fi

1803     if [ "$N_FLAG" != "y" -a -d $SRC/pkg ]; then
1804         echo "\n=== Validating manifests against proto area ===\n" \
1805         >> $mail_msg_file
1806         ( cd $SRC/pkg ; $MAKE -e protocmp ROOT="$checkroot" ) | \
1807         tee $TMPDIR/protocmp_noise >> $mail_msg_file
1808         if [[ -s $TMPDIR/protocmp_noise ]]; then
1809             build_extras_ok=n
1810         fi
1811     fi

1813     if [ "$N_FLAG" != "y" -a -f "$REF_PROTO_LIST" ]; then
1814         echo "\n=== Impact on proto area ===\n" >> $mail_msg_file
1815         if [ -n "$E2" ]; then
1816             ELIST=$E2
1817         else
1818             ELIST=$E1
1819         fi
1820         $PROTCMPERSE \
1821             "Files in yesterday's proto area, but not today's:" \
1822             "Files in today's proto area, but not yesterday's:" \
1823             "Files that changed between yesterday and today:" \
1824             ${ELIST} \
1825             -d $REF_PROTO_LIST \
1826             $ATLOG/proto_list_${MACH} \
1827             >> $mail_msg_file
1828     fi
1829 fi

1831 if [[ "$u_FLAG" == "y" && "$build_ok" == "y" && \
1832     "$build_extras_ok" == "y" ]]; then
1833     staffer cp $ATLOG/proto_list_${MACH} \
1834     $PARENT_WS/usr/src/proto_list_${MACH}
1835 fi

1837 # Update parent proto area if necessary. This is done now
1838 # so that the proto area has either DEBUG or non-DEBUG kernels.
1839 # Note that this clears out the lock file, so we can dispense with

```

```

1840 # the variable now.
1841 if [ "$SU_FLAG" = "y" -a "$build_ok" = "y" ]; then
1842     echo "\n=== Copying proto area to $NIGHTLY_PARENT_ROOT ===\n" | \
1843     tee -a $LOGFILE >> $mail_msg_file
1844     rm -rf $NIGHTLY_PARENT_ROOT/*
1845     unset Ulockfile
1846     mkdir -p $NIGHTLY_PARENT_ROOT
1847     if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
1848         ( cd $ROOT; tar cf - . |
1849         ( cd $NIGHTLY_PARENT_ROOT; umask 0; tar xpf - ) ) 2>&1 |
1850         tee -a $mail_msg_file >> $LOGFILE
1851     fi
1852     if [[ "$MULTI_PROTO" = yes && "$F_FLAG" = n ]]; then
1853         rm -rf $NIGHTLY_PARENT_ROOT-nd/*
1854         mkdir -p $NIGHTLY_PARENT_ROOT-nd
1855         cd $ROOT-nd
1856         ( tar cf - . |
1857         ( cd $NIGHTLY_PARENT_ROOT-nd; umask 0; tar xpf - ) ) 2>&1 |
1858         tee -a $mail_msg_file >> $LOGFILE
1859     fi
1860     if [ -n "${NIGHTLY_PARENT_TOOLS_ROOT}" ]; then
1861         echo "\n=== Copying tools proto area to $NIGHTLY_PARENT_TOOLS_R
1862         tee -a $LOGFILE >> $mail_msg_file
1863         rm -rf $NIGHTLY_PARENT_TOOLS_ROOT/*
1864         mkdir -p $NIGHTLY_PARENT_TOOLS_ROOT
1865         if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
1866             ( cd $TOOLS_PROTO; tar cf - . |
1867             ( cd $NIGHTLY_PARENT_TOOLS_ROOT;
1868             umask 0; tar xpf - ) ) 2>&1 |
1869             tee -a $mail_msg_file >> $LOGFILE
1870         fi
1871     fi
1872 fi

1874 #
1875 # ELF verification: ABI (-A) and runtime (-r) checks
1876 #
1877 if [[ ($build_ok = y) && (($A_FLAG = y) || ($r_FLAG = y)) ]]; then
1878     # Directory ELF-data.$MACH holds the files produced by these tests.
1879     elf_ddir=$SRC/ELF-data.$MACH

1881     # If there is a previous ELF-data backup directory, remove it. Then,
1882     # rotate current ELF-data directory into its place and create a new
1883     # empty directory
1884     rm -rf $elf_ddir.ref
1885     if [[ -d $elf_ddir ]]; then
1886         mv $elf_ddir $elf_ddir.ref
1887     fi
1888     mkdir -p $elf_ddir

1890     # Call find_elf to produce a list of the ELF objects in the proto area.
1891     # This list is passed to check_rtime and interface_check, preventing
1892     # them from separately calling find_elf to do the same work twice.
1893     find_elf -fr $checkroot > $elf_ddir/object_list

1895     if [[ $A_FLAG = y ]]; then
1896         echo "\n=== Check versioning and ABI information ===\n" | \
1897         tee -a $LOGFILE >> $mail_msg_file

1899         # Produce interface description for the proto. Report errors.
1900         interface_check -o -w $elf_ddir -f object_list \
1901             -i interface -E interface.err
1902         if [[ -s $elf_ddir/interface.err ]]; then
1903             tee -a $LOGFILE < $elf_ddir/interface.err \
1904             >> $mail_msg_file
1905             build_extras_ok=n

```

```

1906     fi

1908     # If ELF_DATA_BASELINE_DIR is defined, compare the new interface
1909     # description file to that from the baseline gate. Issue a
1910     # warning if the baseline is not present, and keep going.
1911     if [[ "$ELF_DATA_BASELINE_DIR" != '' ]]; then
1912         base_ifile="$ELF_DATA_BASELINE_DIR/interface"

1914         echo "\n=== Compare versioning and ABI information" \
1915         "to baseline ===\n" | \
1916         tee -a $LOGFILE >> $mail_msg_file
1917         echo "Baseline: $base_ifile\n" >> $LOGFILE

1919         if [[ -f $base_ifile ]]; then
1920             interface_cmp -d -o $base_ifile \
1921                 $elf_ddir/interface > $elf_ddir/interface.cm
1922             if [[ -s $elf_ddir/interface_cmp ]]; then
1923                 echo | tee -a $LOGFILE >> $mail_msg_file
1924                 tee -a $LOGFILE < \
1925                     $elf_ddir/interface_cmp \
1926                     >> $mail_msg_file
1927                 build_extras_ok=n
1928             fi
1929         else
1930             echo "baseline not available. comparison" \
1931                 "skipped" | \
1932                 tee -a $LOGFILE >> $mail_msg_file
1933         fi

1935     fi
1936     fi

1938     if [[ $r_FLAG = y ]]; then
1939         echo "\n=== Check ELF runtime attributes ===\n" | \
1940         tee -a $LOGFILE >> $mail_msg_file

1942         # If we're doing a DEBUG build the proto area will be left
1943         # with debuggable objects, thus don't assert -s.
1944         if [[ $D_FLAG = y ]]; then
1945             rtime_sflag=""
1946         else
1947             rtime_sflag="-s"
1948         fi
1949         check_rtime -i -m -v $rtime_sflag -o -w $elf_ddir \
1950             -D object_list -f object_list -E runtime.err \
1951             -I runtime.attr.raw
1952         if (( $? != 0 )); then
1953             build_extras_ok=n
1954         fi

1956         # check_rtime -I output needs to be sorted in order to
1957         # compare it to that from previous builds.
1958         sort $elf_ddir/runtime.attr.raw > $elf_ddir/runtime.attr
1959         rm $elf_ddir/runtime.attr.raw

1961         # Report errors
1962         if [[ -s $elf_ddir/runtime.err ]]; then
1963             tee -a $LOGFILE < $elf_ddir/runtime.err \
1964             >> $mail_msg_file
1965             build_extras_ok=n
1966         fi

1968         # If there is an ELF-data directory from a previous build,
1969         # then diff the attr files. These files contain information
1970         # about dependencies, versioning, and runpaths. There is some
1971         # overlap with the ABI checking done above, but this also

```

```

1972         # flushes out non-ABI interface differences along with the
1973         # other information.
1974         echo "\n=== Diff ELF runtime attributes" \
1975             "(since last build) ===\n" | \
1976             tee -a $LOGFILE >> $mail_msg_file >> $mail_msg_file

1978         if [[ -f $elf_ddir.ref/runtime.attr ]]; then
1979             diff $elf_ddir.ref/runtime.attr \
1980                 $elf_ddir/runtime.attr \
1981                 >> $mail_msg_file
1982         fi
1983     fi

1985     # If -u set, copy contents of ELF-data.$MACH to the parent workspace.
1986     if [[ "$u_FLAG" = "y" ]]; then
1987         p_elf_ddir=$PARENT_WS/usr/ELF-data.$MACH

1989         # If parent lacks the ELF-data.$MACH directory, create it
1990         if [[ ! -d $p_elf_ddir ]]; then
1991             staffer mkdir -p $p_elf_ddir
1992         fi

1994         # These files are used asynchronously by other builds for ABI
1995         # verification, as above for the -A option. As such, we require
1996         # the file replacement to be atomic. Copy the data to a temp
1997         # file in the same filesystem and then rename into place.
1998         (
1999             cd $elf_ddir
2000             for elf_dfile in *; do
2001                 staffer cp $elf_dfile \
2002                     ${p_elf_ddir}/${elf_dfile}.new
2003                 staffer mv -f ${p_elf_ddir}/${elf_dfile}.new \
2004                     ${p_elf_ddir}/${elf_dfile}
2005             done
2006         )
2007     fi
2008 fi

2010 # DEBUG lint of kernel begins

2012 if [ "$i_CMD_LINE_FLAG" = "n" -a "$l_FLAG" = "y" ]; then
2013     if [ "$LINTDIRS" = "" ]; then
2014         # LINTDIRS="$SRC/uts y $SRC/stand y $SRC/psm y"
2015         LINTDIRS="$SRC y"
2016     fi
2017     set LINTDIRS
2018     while [ $# -gt 0 ]; do
2019         dolint $1 $2; shift; shift
2020     done
2021 else
2022     echo "\n=== No '$MAKE lint' ===\n" >> $LOGFILE
2023 fi

2025 # "make check" begins

2027 if [ "$i_CMD_LINE_FLAG" = "n" -a "$C_FLAG" = "y" ]; then
2028     # remove old check.out
2029     rm -f $SRC/check.out

2031     rm -f $SRC/check-${MACH}.out
2032     cd $SRC
2033     $MAKE -ek check ROOT="$checkroot" 2>&1 | tee -a $SRC/check-${MACH}.out \
2034         >> $LOGFILE
2035     echo "\n=== cstyle/hdrchk errors ===\n" >> $mail_msg_file

2037     grep ":" $SRC/check-${MACH}.out |

```

```

2038         egrep -v "Ignoring unknown host" | \
2039         sort | uniq | tee $TMPDIR/check_errors >> $mail_msg_file

2041     if [[ -s $TMPDIR/check_errors ]]; then
2042         build_extras_ok=n
2043     fi
2044 else
2045     echo "\n=== No '$MAKE check' ===\n" >> $LOGFILE
2046 fi

2048 echo "\n=== Find core files ===\n" | \
2049     tee -a $LOGFILE >> $mail_msg_file

2051 find $abssrkdirs -name core -a -type f -exec file {} \; | \
2052     tee -a $LOGFILE >> $mail_msg_file

2054 if [ "$f_FLAG" = "y" -a "$build_ok" = "y" ]; then
2055     echo "\n=== Diff unreferenced files (since last build) ===\n" \
2056         | tee -a $LOGFILE >> $mail_msg_file
2057     rm -f $SRC/unref-${MACH}.ref
2058     if [ -f $SRC/unref-${MACH}.out ]; then
2059         mv $SRC/unref-${MACH}.out $SRC/unref-${MACH}.ref
2060     fi

2062     findunref -S $SCM_TYPE -t $SRC/.build.tstamp -s usr $CODEMGR_WS \
2063         ${TOOLS}/findunref/exception_list 2>> $mail_msg_file | \
2064         sort > $SRC/unref-${MACH}.out

2066     if [ ! -f $SRC/unref-${MACH}.ref ]; then
2067         cp $SRC/unref-${MACH}.out $SRC/unref-${MACH}.ref
2068     fi

2070     diff $SRC/unref-${MACH}.ref $SRC/unref-${MACH}.out >> $mail_msg_file
2071 fi

2073 # Verify that the usual lists of files, such as exception lists,
2074 # contain only valid references to files. If the build has failed,
2075 # then don't check the proto area.
2076 CHECK_PATHS=${CHECK_PATHS:-y}
2077 if [ "$CHECK_PATHS" = y -a "$N_FLAG" != y ]; then
2078     echo "\n=== Check lists of files ===\n" | tee -a $LOGFILE \
2079         >> $mail_msg_file
2080     arg=-b
2081     [ "$build_ok" = y ] && arg=
2082     checkpaths $arg $checkroot > $SRC/check-paths.out 2>&1
2083     if [[ -s $SRC/check-paths.out ]]; then
2084         tee -a $LOGFILE < $SRC/check-paths.out >> $mail_msg_file
2085     fi
2086     build_extras_ok=n
2087 fi

2089 if [ "$M_FLAG" != "y" -a "$build_ok" = y ]; then
2090     echo "\n=== Impact on file permissions ===\n" \
2091         >> $mail_msg_file

2093     abspkg=
2094     for d in $abssrkdirs; do
2095         if [ -d "$d/pkg" ]; then
2096             abspkg="$abspkg $d"
2097         fi
2098     done

2100     if [ -n "$abspkg" ]; then
2101         for d in "$abspkg"; do
2102             ( cd $d/pkg ; $MAKE -e pmodes ) >> $mail_msg_file
2103         done

```

new/usr/src/tools/scripts/nightly.sh

9

```

2104         fi
2105     fi

2107 if [ "$w_FLAG" = "y" -a "$build_ok" = "y" ]; then
2108     if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
2109         do_wsdiff DEBUG $ROOT.prev $ROOT
2110     fi

2112     if [[ "$MULTI_PROTO" = yes && "$F_FLAG" = n ]]; then
2113         do_wsdiff non-DEBUG $ROOT.nd.prev $ROOT.nd
2114     fi
2115 fi

2117 END_DATE=`date`
2118 echo "==== Nightly $maketype build completed: $END_DATE ==== | \
2119     tee -a $LOGFILE >> $build_time_file

2121 typeset -i10 hours
2122 typeset -Z2 minutes
2123 typeset -Z2 seconds

2125 elapsed_time=$SECONDS
2126 ((hours = elapsed_time / 3600 ))
2127 ((minutes = elapsed_time / 60 % 60))
2128 ((seconds = elapsed_time % 60))

2130 echo "\n==== Total build time ==== | \
2131     tee -a $LOGFILE >> $build_time_file
2132 echo "\nreal    ${hours}:${minutes}:${seconds}" | \
2133     tee -a $LOGFILE >> $build_time_file

2135 if [ "$u_FLAG" = "y" -a "$f_FLAG" = "y" -a "$build_ok" = "y" ]; then
2136     staffer cp ${SRC}/unref-${MACH}.out $PARENT_WS/usr/src/

2138     #
2139     # Produce a master list of unreferenced files -- ideally, we'd
2140     # generate the master just once after all of the nightlyies
2141     # have finished, but there's no simple way to know when that
2142     # will be. Instead, we assume that we're the last nightly to
2143     # finish and merge all of the unref-${MACH}.out files in
2144     # $PARENT_WS/usr/src/. If we are in fact the final ${MACH} to
2145     # finish, then this file will be the authoritative master
2146     # list. Otherwise, another ${MACH}'s nightly will eventually
2147     # overwrite ours with its own master, but in the meantime our
2148     # temporary "master" will be no worse than any older master
2149     # which was already on the parent.
2150     #

2152     set -- $PARENT_WS/usr/src/unref-*.out
2153     cp "$1" ${TMPDIR}/unref.merge
2154     shift

2156     for unreffile; do
2157         comm -12 ${TMPDIR}/unref.merge "$unreffile" > ${TMPDIR}/unref.$$
2158         mv ${TMPDIR}/unref.$$ ${TMPDIR}/unref.merge
2159     done

2161     staffer cp ${TMPDIR}/unref.merge $PARENT_WS/usr/src/unrefmaster.out
2162 fi

2164 #
2165 # All done save for the sweeping up.
2166 # (whichever exit we hit here will trigger the "cleanup" trap which
2167 # optionally sends mail on completion).
2168 #
2169 if [[ "$build_ok" == "y" ]]; then

```

new/usr/src/tools/scripts/nightly.sh

10

```

2170         if [[ "$w_FLAG" == "y" || "$build_extras_ok" == "y" ]]; then
2171             exit 0
2172         fi
2173     fi

2175 exit 1

```