

new/usr/src/Makefile.master

1

```
*****
37959 Wed Apr 24 01:32:49 2019
new/usr/src/Makefile.master
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
27 # Copyright 2015 Gary Mills
28 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
29 # Copyright 2016 Toomas Soome <tsoome@me.com>
30 # Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
31 # Copyright (c) 2019, Joyent, Inc.
32 #
33 #
34 #
35 # Makefile.master, global definitions for system source
36 #
37 ROOT= /proto
38 #
39 #
40 # Adjunct root, containing an additional proto area to be used for headers
41 # and libraries.
42 #
43 ADJUNCT_PROTO=
44 #
45 #
46 # Adjunct for building things that run on the build machine.
47 #
48 NATIVE_ADJUNCT= /usr
49 #
50 #
51 # RELEASE_BUILD should be cleared for final release builds.
52 # NOT_RELEASE_BUILD is exactly what the name implies.
53 #
54 # __GNUC toggles the building of ON components using gcc and related tools.
55 # Normally set to '#', set it to '' to do gcc build.
56 #
57 # The declaration POUND_SIGN is always '#'. This is needed to get around the
58 # make feature that '#' is always a comment delimiter, even when escaped or
59 # quoted. We use this macro expansion method to get POUND_SIGN rather than
```

new/usr/src/Makefile.master

2

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

## new/usr/src/Makefile.master

3

```

126 BLD_JAVA_6=      $(POUND_SIGN)
127 BLD_JAVA_8=      $(POUND_SIGN)

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

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

135 RPCGEN=          /usr/bin/rpcgen
136 STABS=           $(ONBLD_TOOLS)/bin/$(MACH)/stabs
137 ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
138 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
139 BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
140 VTFONTCVT=      $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcv
141 # echo(1) and true(1) are specified without absolute paths, so that the shell
142 # spawned by make(1) may use the built-in versions. This is minimally
143 # problematic, as the shell spawned by make(1) is known and under control, the
144 # only risk being if the shell falls back to $PATH.
145 #
146 # We specifically want an echo(1) that does interpolation of escape sequences,
147 # which ksh93, /bin/sh, and bash will all provide.
148 ECHO=            echo
149 TRUE=            true
150 INS=             $(ONBLD_TOOLS)/bin/$(MACH)/install
151 SYMLINK=         /usr/bin/ln -s
152 LN=              /usr/bin/ln
153 MKDIR=          /usr/bin/mkdir
154 CHMOD=          /usr/bin/chmod
155 MV=              /usr/bin/mv -f
156 RM=              /usr/bin/rm -f
157 CUT=            /usr/bin/cut
158 NM=             /usr/ccs/bin/nm
159 DIFF=           /usr/bin/diff
160 GREP=           /usr/bin/grep
161 EGREP=          /usr/bin/egrep
162 ELFWRAP=        /usr/bin/elfwrap
163 KSH93=          /usr/bin/ksh93
164 SED=            /usr/bin/sed
165 AWK=            /usr/bin/nawk
166 CP=             /usr/bin/cp -f
167 MCS=           /usr/ccs/bin/mcs
168 CAT=           /usr/bin/cat
169 ELFDUMP=        /usr/ccs/bin/elfdump
170 M4=            /usr/bin/m4
171 GM4=           /usr/bin/gm4
172 STRIP=         /usr/ccs/bin/strip
173 LEX=           /usr/ccs/bin/lex
174 FLEX=          /usr/bin/flex
175 YACC=          /usr/ccs/bin/yacc
176 BISON=         /usr/bin/bison
177 CPP=           /usr/lib/cpp
178 ANSI_CPP=      $(GNUC_ROOT)/bin/cpp
179 JAVAC=         $(JAVA_ROOT)/bin/javac
180 JAVAH=         $(JAVA_ROOT)/bin/javah
181 JAVADOC=       $(JAVA_ROOT)/bin/javadoc
182 RMIC=          $(JAVA_ROOT)/bin/rmic
183 JAR=           $(JAVA_ROOT)/bin/jar
184 CTFCONVERT=    $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
185 CTFDIFF=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfdiff
186 CTFMERGE=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
187 CTFSTABS=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
188 CTFSTRIP=      $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
189 NDRGEN=        $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
190 GENOFFSETS=    $(ONBLD_TOOLS)/bin/genoffsets
191 XREF=          $(ONBLD_TOOLS)/bin/xref

```

## new/usr/src/Makefile.master

4

```

192 FIND=          /usr/bin/find
193 PERL=          /usr/bin/perl
194 PERL_VERSION=  5.10.0
195 PERL_PKGVERS=  -510
196 PERL_ARCH =    i86pc-solaris-64int
197 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
198 PYTHON_VERSION= 2.7
199 PYTHON_PKGVERS= -27
200 PYTHON_SUFFIX=
201 PYTHON=         /usr/bin/python$(PYTHON_VERSION)
202 PYTHON3_VERSION= 3.5
203 PYTHON3_PKGVERS= -35
204 PYTHON3_SUFFIX= m
205 PYTHON3=        /usr/bin/python$(PYTHON3_VERSION)
206 $(BUILDPY3TOOLS)TOOLS_PYTHON= $(PYTHON3)
207 $(BUILDPY2TOOLS)TOOLS_PYTHON= $(PYTHON)
208 SORT=           /usr/bin/sort
209 TR=             /usr/bin/tr
210 TOUCH=         /usr/bin/touch
211 WC=            /usr/bin/wc
212 XARGS=         /usr/bin/xargs
213 ELFEDIT=       /usr/bin/elfedit
214 DTRACE=        /usr/sbin/dtrace -xnolib
215 UNIQ=          /usr/bin/uniq
216 TAR=           /usr/bin/tar
217 ASTBINDIR=     /usr/ast/bin
218 MSGCC=         $(ASTBINDIR)/msgcc
219 MSGFMT=        /usr/bin/msgfmt -s
220 LCDEF=         $(ONBLD_TOOLS)/bin/$(MACH)/localedef
221 TIC=           $(ONBLD_TOOLS)/bin/$(MACH)/tic
222 ZIC=           $(ONBLD_TOOLS)/bin/$(MACH)/zic
223 OPENSSL=       /usr/bin/openssl
224 CPCGEN=        $(ONBLD_TOOLS)/bin/$(MACH)/cpcgen

226 FILEMODE=     644
227 DIRMODE=      755

229 # Declare that nothing should be built in parallel.
230 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
231 .NO_PARALLEL:

232 #
233 # For stylistic checks
234 #
235 # Note that the X and C checks are not used at this time and may need
236 # modification when they are actually used.
237 #
238 CSTYLE=        $(ONBLD_TOOLS)/bin/cstyle
239 CSTYLE_TAIL=
240 HDRCHK=        $(ONBLD_TOOLS)/bin/hdrchk
241 HDRCHK_TAIL=
242 JSTYLE=        $(ONBLD_TOOLS)/bin/jstyle

244 DOT_H_CHECK=   \
245   @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL); \
246   $(HDRCHK) $< $(HDRCHK_TAIL)

248 DOT_X_CHECK=   \
249   @$(ECHO) "checking $<"; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
250   $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

252 DOT_C_CHECK=   \
253   @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL)

255 MANIFEST_CHECK= \
256   @$(ECHO) "checking $<"; \
257   SVCCFG_DTD=$(SRC)/cmd/svc/dtd/service_bundle.dtd.1 \

```

```

258 SVCCFG_REPOSITORY=$(SRC)/cmd/svc/seed/global.db \
259 SVCCFG_CONFIGD_PATH=$(SRC)/cmd/svc/configd/svc.configd-native \
260 $(SRC)/cmd/svc/svccfg/svccfg-native validate $<

262 INS.file= $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
263 INS.dir= $(INS) -s -d -m $(DIRMODE) $@
264 # installs and renames at once
265 #
266 INS.rename= $(INS.file); $(MV) $(@D)/$(<F) $@

268 # install a link
269 INSLINKTARGET= $<
270 INS.link= $(RM) $@; $(LN) $(INSLINKTARGET) $@
271 INS.symmlink= $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

273 # The path to python that will be used for the shebang line when installing
274 # python scripts to the proto area. This is overridden by makefiles to
275 # select to the correct version.
276 PYSHEBANG= $(PYTHON)

278 #
279 # Python bakes the mtime of the .py file into the compiled .pyc and
280 # rebuilds if the baked-in mtime != the mtime of the source file
281 # (rather than only if it's less than), thus when installing python
282 # files we must make certain to not adjust the mtime of the source
283 # (.py) file.
284 #
285 INS.pyfile= $(RM) $@; $(SED) \
286 -e "1s:\^#\!@PYTHON@:\#\!$(PYSHEBANG):" \
287 -e "1s:\^#\!@TOOLS_PYTHON@:\#\!$(TOOLS_PYTHON):" \
288 < $< > $@; $(CHMOD) $(FILEMODE) $@; $(TOUCH) -r $< $@

290 # MACH must be set in the shell environment per uname -p on the build host
291 # More specific architecture variables should be set in lower makefiles.
292 #
293 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
294 # architectures on which we do not build 64-bit versions.
295 # (There are no such architectures at the moment.)
296 #
297 # Set BUILD64=# in the environment to disable 64-bit amd64
298 # builds on i386 machines.

300 MACH64_1= $(MACH:sparc=sparcv9)
301 MACH64= $(MACH64_1:i386=amd64)

303 MACH32_1= $(MACH:sparc=sparcv7)
304 MACH32= $(MACH32_1:i386=i86)

306 sparc_BUILD64=
307 i386_BUILD64=
308 BUILD64= $(($(MACH)_BUILD64))

310 #
311 # C compiler mode. Future compilers may change the default on us,
312 # so force extended ANSI mode globally. Lower level makefiles can
313 # override this by setting CCMODE.
314 #
315 CCMODE= -Xa
316 CCMODE64= -Xa

318 #
319 # C compiler verbose mode. This is so we can enable it globally,
320 # but turn it off in the lower level makefiles of things we cannot
321 # (or aren't going to) fix.
322 #
323 CCVERBOSE= -v

```

```

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

329 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
330 # symbols (used to detect conflicts between objects that use global registers)
331 # we disable this now for safety, and because genunix doesn't link with
332 # this feature (the v9 default) enabled.
333 #
334 # REGSYM is separate since the C++ driver syntax is different.
335 CCREGSYM= -Wc,-Qiselect-regsym=0
336 CCCREGSYM= -Qoption cg -Qiselect-regsym=0

338 # Prevent the removal of static symbols by the SPARC code generator (cg).
339 # The x86 code generator (ube) does not remove such symbols and as such
340 # using this workaround is not applicable for x86.
341 #
342 CCSTATICSYM= -Wc,-Qassembler-ounrefsym=0
343 #
344 # generate 32-bit addresses in the v9 kernel. Saves memory.
345 CCABS32= -Wc,-xcode=abs32
346 #
347 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
348 # system calls.
349 CC32BITCALLERS= -_gcc=-massume-32bit-callers

351 # GCC, especially, is increasingly beginning to auto-inline functions and
352 # sadly does so separately not under the general -fno-inline-functions
353 # Additionally, we wish to prevent optimisations which cause GCC to clone
354 # functions -- in particular, these may cause unhelpful symbols to be
355 # emitted instead of function names
356 CCNOAUTOINLINE= \
357 -_gcc=-fno-inline-small-functions \
358 -_gcc=-fno-inline-functions-called-once \
359 -_gcc=-fno-ipa-cp \
360 -_gcc7=-fno-ipa-icf \
361 -_gcc8=-fno-ipa-icf \
362 -_gcc7=-fno-clone-functions \
363 -_gcc8=-fno-clone-functions

365 # GCC may put functions in different named sub-sections of .text based on
366 # their presumed calling frequency. At least in the kernel, where we actually
367 # deliver relocatable objects, we don't want this to happen.
368 #
369 # Since at present we don't benefit from this even in userland, we disable it gl
370 # but the application of this may move into usr/src/uts/ in future.
371 CCNOREORDER= \
372 -_gcc7=-fno-reorder-functions \
373 -_gcc8=-fno-reorder-functions

375 #
376 # gcc has a rather aggressive optimization on by default that infers loop
377 # bounds based on undefined behavior (!!). This can lead to some VERY
378 # surprising optimizations -- ones that may be technically correct in the
379 # strictest sense but also result in incorrect program behavior. We turn
380 # this optimization off, with extreme prejudice.
381 #
382 CCNOAGGRESSIVELOOPS= \
383 -_gcc7=-fno-aggressive-loop-optimizations \
384 -_gcc8=-fno-aggressive-loop-optimizations

386 # One optimization the compiler might perform is to turn this:
387 # #pragma weak foo
388 # extern int foo;
389 # if (&foo)

```

```

390 #         foo = 5;
391 # into
392 #         foo = 5;
393 # Since we do some of this (foo might be referenced in common kernel code
394 # but provided only for some cpu modules or platforms), we disable this
395 # optimization.
396 #
397 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
398 i386_CCUNBOUND =
399 CCUNBOUND = $($ (MACH)_CCUNBOUND)

401 #
402 # compiler '-xarch' flag. This is here to centralize it and make it
403 # overridable for testing.
404 sparc_XARCH=      -m32
405 sparcv9_XARCH=   -m64
406 i386_XARCH=      -m32
407 amd64_XARCH=     -m64 -Ui386 -U__i386

409 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
410 sparc_AS_XARCH=   -xarch=v8plus
411 sparcv9_AS_XARCH= -xarch=v9
412 i386_AS_XARCH=
413 amd64_AS_XARCH=   -xarch=amd64 -P -Ui386 -U__i386

415 #
416 # These flags define what we need to be 'standalone' i.e. -not- part
417 # of the rather more cosy userland environment. This basically means
418 # the kernel.
419 #
420 # XX64 future versions of gcc will make -mmodel=kernel imply -mno-red-zone
421 #
422 sparc_STAND_FLAGS= -_gcc=-ffreestanding
423 sparcv9_STAND_FLAGS= -_gcc=-ffreestanding
424 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
425 # additions to SSE (SSE2, AVX ,etc.)
426 NO_SIMD=          -_gcc=-mno-mmx -_gcc=-mno-sse
427 i386_STAND_FLAGS= -_gcc=-ffreestanding $(NO_SIMD)
428 amd64_STAND_FLAGS= -xmodel=kernel $(NO_SIMD)

430 SAVEARGS=         -Wu,-save_args
431 amd64_STAND_FLAGS += $(SAVEARGS)

433 STAND_FLAGS_32 = $($ (MACH)_STAND_FLAGS)
434 STAND_FLAGS_64 = $($ (MACH64)_STAND_FLAGS)

436 #
437 # disable the incremental linker
438 ILDOFF=           -xildoff
439 #
440 XFFLAG=          -xF=%all
441 XESS=           -xs
442 XSTRCONST=      -xstrconst

444 #
445 # turn warnings into errors (C)
446 CERRWARN = -errtags=yes -errwarn=%all
447 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
448 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

450 CERRWARN += -_gcc=-Wno-missing-braces
451 CERRWARN += -_gcc=-Wno-sign-compare
452 CERRWARN += -_gcc=-Wno-unknown-pragmas
453 CERRWARN += -_gcc=-Wno-unused-parameter
454 CERRWARN += -_gcc=-Wno-missing-field-initializers

```

```

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

459 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
460 # -nd builds
461 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
462 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

460 CERRWARN += -_smatch=-p=illumos_user
461 include $(SRC)/Makefile.smatch

463 #
464 # turn warnings into errors (C++)
465 CCERRWARN= -xwe

467 # C standard. Keep Studio flags until we get rid of lint.
468 CSTD_GNU89= -xc99=%none
469 CSTD_GNU99= -xc99=%all
470 CSTD= $(CSTD_GNU89)
471 C99LMODE= $(CSTD:-xc99%=-Xc99%)

473 # In most places, assignments to these macros should be appended with +=
474 # (CPPFLAGS.first allows values to be prepended to CPPFLAGS).
475 sparc_CFLAGS= $(sparc_XARCH) $(CCSTATICSYM)
476 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
477 $(CCSTATICSYM)
478 i386_CFLAGS= $(i386_XARCH)
479 amd64_CFLAGS= $(amd64_XARCH)

481 sparc_ASFLAGS= $(sparc_AS_XARCH)
482 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
483 i386_ASFLAGS= $(i386_AS_XARCH)
484 amd64_ASFLAGS= $(amd64_AS_XARCH)

486 #
487 sparc_COPTFLAG= -xO3
488 sparcv9_COPTFLAG= -xO3
489 i386_COPTFLAG= -O
490 amd64_COPTFLAG= -xO3

492 COPTFLAG= $($ (MACH)_COPTFLAG)
493 COPTFLAG64= $($ (MACH64)_COPTFLAG)

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

499 # Direct the Sun Studio compiler to use a static globalization prefix based on t
500 # name of the module rather than something unique. Otherwise, objects
501 # will not build deterministically, as subsequent compilations of identical
502 # source will yield objects that always look different.
503 #
504 # In the same spirit, this will also remove the date from the N_OPT stab.
505 CGLOBALSTATIC= -W0,-xglobalstatic

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

511 #
512 # We force the compilers to generate the debugging information best understood
513 # by the CTF tools. With Sun Studio this is stabs due to bugs in the Studio
514 # compilers. With GCC this is DWARF v2.
515 #
516 DEBUGFORMAT= -_cc=-xdebugformat=stabs -_gcc=-gdwarf-2

```

```

518 #
519 # Ask the compiler to include debugging information
520 #
521 CCGDEBUG= -g $(DEBUGFORMAT)

523 #
524 # Flags used to build in debug mode for ctf generation.
525 #
526 CTF_FLAGS_sparc = $(CCGDEBUG) -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL)
527 CTF_FLAGS_i386 = $(CCGDEBUG) $(CSTD) $(CNOGLOBAL)

529 CTF_FLAGS_sparcv9 = $(CTF_FLAGS_sparc)
530 CTF_FLAGS_amd64 = $(CTF_FLAGS_i386)

532 # Sun Studio produces broken userland code when saving arguments.
533 $(__GNUCC)CTF_FLAGS_amd64 += $(SAVEARGS)

535 CTF_FLAGS_32 = $(CTF_FLAGS_$(MACH))
536 CTF_FLAGS_64 = $(CTF_FLAGS_$(MACH64))
537 CTF_FLAGS = $(CTF_FLAGS_32)

539 #
540 # Flags used with genoffsets
541 #
542 GENOFFSETS_FLAGS = $(CALLSYMS)

544 OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
545 $(CW) --noecho $(CW_CC_COMPILERS) -- $(GENOFFSETS_FLAGS) \
546 $(CFLAGS) $(CPPFLAGS)

548 OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
549 $(CW) --noecho $(CW_CC_COMPILERS) -- $(GENOFFSETS_FLAGS) \
550 $(CFLAGS64) $(CPPFLAGS)

552 #
553 # tradeoff time for space (smaller is better)
554 #
555 sparc_SPACEFLAG = -xspace -W0,-Lt
556 sparcv9_SPACEFLAG = -xspace -W0,-Lt
557 i386_SPACEFLAG = -xspace
558 amd64_SPACEFLAG =

560 SPACEFLAG = $(($(MACH)_SPACEFLAG)
561 SPACEFLAG64 = $(($(MACH64)_SPACEFLAG)

563 #
564 # The Sun Studio 11 compiler has changed the behaviour of integer
565 # wrap arounds and so a flag is needed to use the legacy behaviour
566 # (without this flag panics/hangs could be exposed within the source).
567 #
568 sparc_IROPTFLAG = -W2,-xwrap_int
569 sparcv9_IROPTFLAG = -W2,-xwrap_int
570 i386_IROPTFLAG =
571 amd64_IROPTFLAG =

573 IROPTFLAG = $(($(MACH)_IROPTFLAG)
574 IROPTFLAG64 = $(($(MACH64)_IROPTFLAG)

576 sparc_XREGSFLAG = -xregs=no%appl
577 sparcv9_XREGSFLAG = -xregs=no%appl
578 i386_XREGSFLAG =
579 amd64_XREGSFLAG =

581 XREGSFLAG = $(($(MACH)_XREGSFLAG)
582 XREGSFLAG64 = $(($(MACH64)_XREGSFLAG)

```

```

584 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
585 # avoids stripping it.
586 SOURCEDEBUG = $(POUND_SIGN)
587 SRCDBGBLD = $(SOURCEDEBUG:yes=)

589 #
590 # These variables are intended ONLY for use by developers to safely pass extra
591 # flags to the compilers without unintentionally overriding Makefile-set
592 # flags. They should NEVER be set to any value in a Makefile.
593 #
594 # They come last in the associated FLAGS variable such that they can
595 # explicitly override things if necessary, there are gaps in this, but it's
596 # the best we can manage.
597 #
598 CUSERFLAGS =
599 CUSERFLAGS64 = $(CUSERFLAGS)
600 CCUSERFLAGS =
601 CCUSERFLAGS64 = $(CCUSERFLAGS)

603 CSOURCEDEBUGFLAGS =
604 CCSOURCEDEBUGFLAGS =
605 $(SRCDBGBLD)CSOURCEDEBUGFLAGS = $(CCGDEBUG) -xs
606 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS = $(CCGDEBUG) -xs

608 CFLAGS= $(COPTFLAG) $(($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
609 $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG) \
610 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CCNOREORDER) \
611 $(CCNOAGGRESSIVELOOPS) \
612 $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)
613 CFLAGS64= $(COPTFLAG64) $(($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
614 $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG64) \
615 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CCNOREORDER) \
616 $(CCNOAGGRESSIVELOOPS) \
617 $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS64)
618 #
619 # Flags that are used to build parts of the code that are subsequently
620 # run on the build machine (also known as the NATIVE_BUILD).
621 #
622 NATIVE_CFLAGS= $(COPTFLAG) $(($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
623 $(ILDOFF) $(CERRWARN) $(CSTD) $(($(NATIVE_MACH)_CCUNBOUND) \
624 $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
625 $(CCNOREORDER) $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

627 DTEXTDOM=-DTEXT_DOMAIN="\$(TEXT_DOMAIN)\\" # For messaging.
628 DTS_ERRNO=-D_TS_ERRNO
629 CPPFLAGS.first= # Please keep empty. Only lower makefiles should set this.
630 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
631 $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
632 $(ADJUNCT_PROTO:%=-I%/usr/include)
633 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
634 $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
635 CPPFLAGS= $(CPPFLAGS.first) $(CPPFLAGS.master)
636 AS_CPPFLAGS= $(CPPFLAGS.first) $(CPPFLAGS.master)
637 JAVAFLAGS= -source 1.6 -target 1.6 -Xlint:deprecation,-options

639 #
640 # For source message catalogue
641 #
642 .SUFFIXES: $(SUFFIXES) .i .po
643 MSGROOT= $(ROOT)/catalog
644 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
645 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
646 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
647 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

649 CLOBBERFILES += $(POFILE) $(POFILES)

```

```

650 COMPILER.cpp= $(CC) -E -C $(CFLAGS) $(CPPFLAGS)
651 XGETTEXT= /usr/bin/xgettext
652 XGETTEXTFLAGS= -c TRANSLATION_NOTE
653 GNUXGETTEXT= /usr/gnu/bin/xgettext
654 GNUXGETTEXTFLAGS= --add-comments=TRANSLATION_NOTE --keyword=_ \
655 --strict --no-location --omit-header
656 BUILD.po= $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $<.i ;\
657 $(RM) $@ ;\
658 $(SED) "/^domain/d" < $(<F).po > $@ ;\
659 $(RM) $(<F).po $<.i

661 #
662 # This is overwritten by local Makefile when PROG is a list.
663 #
664 POFILE= $(PROG).po

666 sparc_CCFLAGS= -cg92 -compat=4 \
667 -Option ccfe -messages=no%anachronism \
668 $(CCERRWARN)
669 sparcv9_CCFLAGS= $(sparcv9_XARCH) -dalign -compat=5 \
670 -Option ccfe -messages=no%anachronism \
671 -Option ccfe -features=no%conststrings \
672 $(CCREGSYM) \
673 $(CCERRWARN)
674 i386_CCFLAGS= -compat=4 \
675 -Option ccfe -messages=no%anachronism \
676 -Option ccfe -features=no%conststrings \
677 $(CCERRWARN)
678 amd64_CCFLAGS= $(amd64_XARCH) -compat=5 \
679 -Option ccfe -messages=no%anachronism \
680 -Option ccfe -features=no%conststrings \
681 $(CCERRWARN)

683 sparc_CCOPTFLAG= -O
684 sparcv9_CCOPTFLAG= -O
685 i386_CCOPTFLAG= -O
686 amd64_CCOPTFLAG= -O

688 CCOPTFLAG= $($ (MACH)_CCOPTFLAG)
689 CCOPTFLAG64= $($ (MACH64)_CCOPTFLAG)
690 CCFLAGS= $(CCOPTFLAG) $($ (MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
691 $(CCUSERFLAGS)
692 CCFLAGS64= $(CCOPTFLAG64) $($ (MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
693 $(CCUSERFLAGS64)

695 #
696 #
697 #
698 ELFWRAP_FLAGS =
699 ELFWRAP_FLAGS64 = -64

701 #
702 # Various mapfiles that are used throughout the build, and delivered to
703 # /usr/lib/ld.
704 #
705 MAPFILE.NED_i386 = $(SRC)/common/mapfiles/common/map.noexdata
706 MAPFILE.NED_sparc =
707 MAPFILE.NED = $(MAPFILE.NED_$(MACH))
708 MAPFILE.PGA = $(SRC)/common/mapfiles/common/map.pagealign
709 MAPFILE.NES = $(SRC)/common/mapfiles/common/map.noexstk
710 MAPFILE.FLT = $(SRC)/common/mapfiles/common/map.filter
711 MAPFILE.LEX = $(SRC)/common/mapfiles/common/map.lex.yy

713 #
714 # Generated mapfiles that are compiler specific, and used throughout the
715 # build. These mapfiles are not delivered in /usr/lib/ld.

```

```

716 #
717 MAPFILE.NGB_sparc= $(SRC)/common/mapfiles/gen/sparc_cc_map.noexglobs
718 $(__GNUC64)MAPFILE.NGB_sparc= \
719 $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexglobs
720 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexglobs
721 $(__GNUC64)MAPFILE.NGB_sparcv9= \
722 $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexglobs
723 MAPFILE.NGB_i386= $(SRC)/common/mapfiles/gen/i386_cc_map.noexglobs
724 $(__GNUC64)MAPFILE.NGB_i386= \
725 $(SRC)/common/mapfiles/gen/i386_gcc_map.noexglobs
726 MAPFILE.NGB_amd64= $(SRC)/common/mapfiles/gen/amd64_cc_map.noexglobs
727 $(__GNUC64)MAPFILE.NGB_amd64= \
728 $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexglobs
729 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH))

731 #
732 # A generic interface mapfile name, used by various dynamic objects to define
733 # the interfaces and interposers the object must export.
734 #
735 MAPFILE.INT = mapfile-intf

737 #
738 # LDLIBS32 and LDLIBS64 can be set in the environment to override the following
739 # assignments.
740 #
741 # These environment settings make sure that no libraries are searched outside
742 # of the local workspace proto area:
743 # LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
744 # LDLIBS64=-YP,$ROOT/lib:$MACH64:$ROOT/usr/lib:$MACH64
745 #
746 LDLIBS32 = $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
747 LDLIBS32 += $(ADJUNCT_PROTO:%=-L%/usr/lib -L%/lib)
748 LDLIBS.cmd = $(LDLIBS32)
749 LDLIBS.lib = $(LDLIBS32)

751 LDLIBS64 = $(ENVLDLIBS1:%=%/(MACH64)) \
752 $(ENVLDLIBS2:%=%/(MACH64)) \
753 $(ENVLDLIBS3:%=%/(MACH64))
754 LDLIBS64 += $(ADJUNCT_PROTO:%=-L%/usr/lib/(MACH64) -L%/lib/(MACH64))

756 #
757 # Define compilation macros.
758 #
759 COMPILER.c= $(CC) $(CFLAGS) $(CPPFLAGS) -c
760 COMPILER64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) -c
761 COMPILER.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
762 COMPILER64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
763 COMPILER.s= $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
764 COMPILER64.s= $(AS) $(ASFLAGS) $(MACH64)_AS_XARCH $(AS_CPPFLAGS)
765 COMPILER.d= $(DTRACE) -G -32
766 COMPILER64.d= $(DTRACE) -G -64
767 COMPILER.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
768 COMPILER64.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

770 CLASSPATH= .
771 COMPILER.java= $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

773 #
774 # Link time macros
775 #
776 CCNEEDED = -lc
777 CCEXTNEEDED = -lc_r -lcstd
778 $(__GNUC)CCNEEDED = -L$(GCCLIBDIR) -lstc++ -lgcc_s
779 $(__GNUC)CCEXTNEEDED = $(CCNEEDED)

781 LINK.c= $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)

```

```

782 LINK64.c=      $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
783 NORUNPATH=     -norunpath -nolib
784 LINK.cc=       $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
785               $(LDFLAGS) $(CCNEEDED)
786 LINK64.cc=     $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
787               $(LDFLAGS) $(CCNEEDED)

789 #
790 # lint macros
791 #
792 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
793 # ON is built with a version of lint that has the fix for 4484186.
794 #
795 ALWAYS_LINT_DEFS =      -errtags=yes -s
796 ALWAYS_LINT_DEFS +=    -erroff=E_PTRDIFF_OVERFLOW
797 ALWAYS_LINT_DEFS +=    -erroff=E_ASSIGN_NARROW_CONV
798 ALWAYS_LINT_DEFS +=    -U__PRAGMA_REDEFINE_EXTNAME
799 ALWAYS_LINT_DEFS +=    $(C99LMODE)
800 ALWAYS_LINT_DEFS +=    -errsecurity=$(SECLEVEL)
801 ALWAYS_LINT_DEFS +=    -erroff=E_SEC_CREAT_WITHOUT_EXCL
802 ALWAYS_LINT_DEFS +=    -erroff=E_SEC_FORBIDDEN_WARN_CREAT
803 # XX64 -- really only needed for amd64 lint
804 ALWAYS_LINT_DEFS +=    -erroff=E_ASSIGN_INT_TO_SMALL_INT
805 ALWAYS_LINT_DEFS +=    -erroff=E_CAST_INT_CONST_TO_SMALL_INT
806 ALWAYS_LINT_DEFS +=    -erroff=E_CAST_INT_TO_SMALL_INT
807 ALWAYS_LINT_DEFS +=    -erroff=E_CAST_TO_PTR_FROM_INT
808 ALWAYS_LINT_DEFS +=    -erroff=E_COMP_INT_WITH_LARGE_INT
809 ALWAYS_LINT_DEFS +=    -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
810 ALWAYS_LINT_DEFS +=    -erroff=E_PASS_INT_TO_SMALL_INT
811 ALWAYS_LINT_DEFS +=    -erroff=E_PTR_CONV_LOSES_BITS

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

817 SECLEVEL=         core
818 LINT.c=            $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
819                   $(ALWAYS_LINT_DEFS)
820 LINT64.c=          $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
821                   $(ALWAYS_LINT_DEFS)
822 LINT.s=            $(LINT.c)

824 # For some future builds, NATIVE_MACH and MACH might be different.
825 # Therefore, NATIVE_MACH needs to be redefined in the
826 # environment as 'uname -p' to override this macro.
827 #
828 # For now at least, we cross-compile amd64 on i386 machines.
829 NATIVE_MACH=       $(MACH:amd64=i386)

831 # Define native compilation macros
832 #

834 # Base directory where compilers are loaded.
835 # Defined here so it can be overridden by developer.
836 #
837 SPRO_ROOT=        $(BUILD_TOOLS)/SUNwspro
838 SPRO_VROOT=       $(SPRO_ROOT)/SS12
839 GNU_ROOT=         /usr

841 $(__GNUC)PRIMARY_CC= gcc4,$(GNUC_ROOT)/bin/gcc,gnu
842 $(__SUNC)PRIMARY_CC= studio12,$(SPRO_VROOT)/bin/cc,sun
843 $(__GNUC)PRIMARY_CCC= gcc4,$(GNUC_ROOT)/bin/g++,gnu
844 $(__SUNC)PRIMARY_CCC= studio12,$(SPRO_VROOT)/bin/CC,sun

846 CW_CC_COMPILERS=  $(PRIMARY_CC:%=--primary %) $(SHADOW_CCS:%=--shadow %)
847 CW_CCC_COMPILERS= $(PRIMARY_CCC:%=--primary %) $(SHADOW_CCCS:%=--shadow %)

```

```

850 # Till SS12ul formally becomes the NV CBE, LINT is hard
851 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
852 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
853 # i386_LINT, amd64_LINT.
854 # Reset them when SS12ul is rolled out.
855 #

857 # Specify platform compiler versions for languages
858 # that we use (currently only c and c++).
859 #
860 CW=                $(ONBLD_TOOLS)/bin/$(MACH)/cw

862 BUILD_CC=         $(CW) $(CW_CC_COMPILERS) --
863 BUILD_CCC=        $(CW) -C $(CW_CCC_COMPILERS) --
864 BUILD_CPP=        /usr/ccs/lib/cpp
865 BUILD_LD=         /usr/ccs/bin/ld
866 BUILD_LINT=       $(SPRO_ROOT)/sunstudio12.1/bin/lint

868 $(MACH)_CC=       $(BUILD_CC)
869 $(MACH)_CCC=      $(BUILD_CCC)
870 $(MACH)_CPP=      $(BUILD_CPP)
871 $(MACH)_LD=       $(BUILD_LD)
872 $(MACH)_LINT=     $(BUILD_LINT)
873 $(MACH64)_CC=     $(BUILD_CC)
874 $(MACH64)_CCC=   $(BUILD_CCC)
875 $(MACH64)_CPP=   $(BUILD_CPP)
876 $(MACH64)_LD=    $(BUILD_LD)
877 $(MACH64)_LINT=  $(BUILD_LINT)

879 sparc_AS=         /usr/ccs/bin/as -xregsym=no
880 sparcv9_AS=       $(MACH)_AS

882 i386_AS=          /usr/ccs/bin/as
883 $(__GNUC)i386_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
884 amd64_AS=         $(ONBLD_TOOLS)/bin/$(MACH)/aw

886 NATIVECC=        $(MACH)_CC
887 NATIVECCC=       $(MACH)_CCC
888 NATIVECPP=       $(MACH)_CPP
889 NATIVEAS=        $(MACH)_AS
890 NATIVELD=        $(MACH)_LD
891 NATIVELINT=      $(MACH)_LINT

893 #
894 # Makefile.master.64 overrides these settings
895 #
896 CC=               $(NATIVECC)
897 CCC=              $(NATIVECCC)
898 CPP=              $(NATIVECPP)
899 AS=               $(NATIVEAS)
900 LD=               $(NATIVELD)
901 LINT=             $(NATIVELINT)

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

906 BDIRECT=         -Bdirect
907 BDYNAMIC=        -Bdynamic
908 BLOCAL=          -Blocal
909 BNODIRECT=       -Bnodirect
910 BREDUCE=         -Breduce
911 BSTATIC=         -Bstatic

913 ZDEFS=           -zdefs

```

```

914 ZDIRECT=      -zdirect
915 ZIGNORE=      -zignore
916 ZINITFIRST=   -zinitfirst
917 ZINTERPOSE=   -zinterpose
918 ZLAZYLOAD=    -zlazyload
919 ZLOADFLTR=    -zloadfltr
920 ZMULDEFS=     -zmuldefs
921 ZNODEFAULTLIB=-znodefaultlib
922 ZNODEFS=      -znodefs
923 ZNODELETE=    -znodelete
924 ZNODLOPEN=    -znodlopen
925 ZNODUMP=      -znodump
926 ZNOLAZYLOAD=  -znolazyload
927 ZNOLDYNSYM=  -znoldynsym
928 ZNORELOC=     -znoreloc
929 ZNOVERSION=   -znoverion
930 ZRECORD=      -zrecord
931 ZREDLOCSYM=   -zredlocsym
932 ZTEXT=        -ztext
933 ZVERBOSE=     -zverbose

935 GSHARED=      -G
936 CCMT=         -mt

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

941 sparc_C_PICFLAGS =      -fpic
942 sparcv9_C_PICFLAGS =    -fpic
943 i386_C_PICFLAGS =       -fpic
944 amd64_C_PICFLAGS =      -fpic
945 C_PICFLAGS =            $(($(MACH)_C_PICFLAGS))
946 C_PICFLAGS64 =          $(($(MACH64)_C_PICFLAGS))

948 sparc_C_BIGPICFLAGS =   -fPIC
949 sparcv9_C_BIGPICFLAGS = -fPIC
950 i386_C_BIGPICFLAGS =    -fPIC
951 amd64_C_BIGPICFLAGS =   -fPIC
952 C_BIGPICFLAGS =        $(($(MACH)_C_BIGPICFLAGS))
953 C_BIGPICFLAGS64 =      $(($(MACH64)_C_BIGPICFLAGS))

955 # CC requires there to be no space between '-K' and 'pic' or 'PIC'.
956 # and does not support -f
957 sparc_CC_PICFLAGS =      -_cc=-Kpic -_gcc=-fpic
958 sparcv9_CC_PICFLAGS =    -_cc=-KPIC -_gcc=-fPIC
959 i386_CC_PICFLAGS =       -_cc=-Kpic -_gcc=-fpic
960 amd64_CC_PICFLAGS =      -_cc=-Kpic -_gcc=-fpic
961 CC_PICFLAGS =            $(($(MACH)_CC_PICFLAGS))
962 CC_PICFLAGS64 =          $(($(MACH64)_CC_PICFLAGS))

964 AS_PICFLAGS=            -K pic
965 AS_BIGPICFLAGS=         -K PIC

967 #
968 # Default label for CTF sections
969 #
970 CTFCVTFLAGS=            -i -L VERSION

972 #
973 # Override to pass module-specific flags to ctfmerge. Currently used only by
974 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
975 # stripping.
976 #
977 CTFMRGFLAGS=

979 #

```

```

980 # Make the transition between old and new CTF Tools. The new ctf tools
981 # do not support stabs (eg. Sun Studio). By setting BUILD_OLD_CTF_TOOLS
982 # here or in the environment file, the old ones will be built.
983 #
984 BUILD_NEW_CTF_TOOLS=
985 BUILD_OLD_CTF_TOOLS=$(POUND_SIGN)
986 $(BUILD_OLD_CTF_TOOLS)BUILD_NEW_CTF_TOOLS= $(POUND_SIGN)

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

990 # Rules (normally from make.rules) and macros which are used for post
991 # processing files. Normally, these do stripping of the comment section
992 # automatically.
993 #   RELEASE_CM:          Should be edited to reflect the release.
994 #   POST_PROCESS_O:     Post-processing for '.o' files.
995 #   POST_PROCESS_A:     Post-processing for '.a' files (currently null).
996 #   POST_PROCESS_SO:    Post-processing for '.so' files.
997 #   POST_PROCESS:       Post-processing for executable files (no suffix).
998 # Note that these macros are not completely generalized as they are to be
999 # used with the file name to be processed following.
1000 #
1001 # It is left as an exercise to Release Engineering to embellish the generation
1002 # of the release comment string.
1003 #
1004 #   If this is a standard development build:
1005 #       compress the comment section (mcs -c)
1006 #       add the standard comment (mcs -a $(RELEASE_CM))
1007 #       add the development specific comment (mcs -a $(DEV_CM))
1008 #
1009 #   If this is an installation build:
1010 #       delete the comment section (mcs -d)
1011 #       add the standard comment (mcs -a $(RELEASE_CM))
1012 #       add the development specific comment (mcs -a $(DEV_CM))
1013 #
1014 #   If this is an release build:
1015 #       delete the comment section (mcs -d)
1016 #       add the standard comment (mcs -a $(RELEASE_CM))
1017 #
1018 # The following list of macros are used in the definition of RELEASE_CM
1019 # which is used to label all binaries in the build:
1020 #
1021 #   RELEASE             Specific release of the build, eg: 5.2
1022 #   RELEASE_MAJOR       Major version number part of $(RELEASE)
1023 #   RELEASE_MINOR       Minor version number part of $(RELEASE)
1024 #   VERSION             Version of the build (alpha, beta, Generic)
1025 #   PATCHID             If this is a patch this value should contain
1026 #                       the patchid value (eg: "Generic 100832-01"), otherwise
1027 #                       it will be set to $(VERSION)
1028 #   RELEASE_DATE        Date of the Release Build
1029 #   PATCH_DATE          Date the patch was created, if this is blank it
1030 #                       will default to the RELEASE_DATE
1031 #
1032 RELEASE_MAJOR= 5
1033 RELEASE_MINOR= 11
1034 RELEASE=       $(RELEASE_MAJOR).$(RELEASE_MINOR)
1035 VERSION=       SunOS Development
1036 PATCHID=      $(VERSION)
1037 RELEASE_DATE= release date not set
1038 PATCH_DATE=   $(RELEASE_DATE)
1039 RELEASE_CM=   "@$(POUND_SIGN)SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
1040 DEV_CM=       "@$(POUND_SIGN)SunOS Internal Development: non-nightly build"

1042 PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
1043 $(RELEASE_BUILD)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM)

1045 STRIP_STABS=   $(STRIP) -x $@

```



```

1046 $(SRCDBGBLD)STRIP_STABS=      :
1048 POST_PROCESS_O=
1049 POST_PROCESS_A=
1050 POST_PROCESS_SO=                $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
1051                                $(ELFSIGN_OBJECT)
1052 POST_PROCESS=                   $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
1053                                $(ELFSIGN_OBJECT)

1055 #
1056 # chk4ubin is a tool that inspects a module for a symbol table
1057 # ELF section size which can trigger an OBP bug on older platforms.
1058 # This problem affects only specific sun4u bootable modules.
1059 #
1060 CHK4UBIN=                        $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
1061 CHK4UBINFLAGS=
1062 CHK4UBINARY=                     $(CHK4UBIN) $(CHK4UBINFLAGS) $@

1064 #
1065 # PKGARCHIVE specifies the default location where packages should be
1066 # placed if built.
1067 #
1068 $(RELEASE_BUILD)PKGARCHIVESUFFIX= -nd
1069 PKGARCHIVE=$(SRC)/../../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)

1071 #
1072 # The repositories will be created with these publisher settings. To
1073 # update an image to the resulting repositories, this must match the
1074 # publisher name provided to "pkg set-publisher."
1075 #
1076 PKGPUBLISHER_REDIST=             on-nightly
1077 PKGPUBLISHER_NONREDIST=          on-extra

1079 #           Default build rules which perform comment section post-processing.
1080 #
1081 .c:
1082     $(LINK.c) -o $@ $< $(LDLIBS)
1083     $(POST_PROCESS)
1084 .c.o:
1085     $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1086     $(POST_PROCESS_O)
1087 .c.a:
1088     $(COMPILE.c) -o $% $<
1089     $(PROCESS_COMMENT) $%
1090     $(AR) $(ARFLAGS) $@ $%
1091     $(RM) $%
1092 .s.o:
1093     $(COMPILE.s) -o $@ $<
1094     $(POST_PROCESS_O)
1095 .s.a:
1096     $(COMPILE.s) -o $% $<
1097     $(PROCESS_COMMENT) $%
1098     $(AR) $(ARFLAGS) $@ $%
1099     $(RM) $%
1100 .cc:
1101     $(LINK.cc) -o $@ $< $(LDLIBS)
1102     $(POST_PROCESS)
1103 .cc.o:
1104     $(COMPILE.cc) $(OUTPUT_OPTION) $<
1105     $(POST_PROCESS_O)
1106 .cc.a:
1107     $(COMPILE.cc) -o $% $<
1108     $(AR) $(ARFLAGS) $@ $%
1109     $(PROCESS_COMMENT) $%
1110     $(RM) $%
1111 .y:

```

```

1112     $(YACC.y) $<
1113     $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1114     $(POST_PROCESS)
1115     $(RM) y.tab.c
1116 .y.o:
1117     $(YACC.y) $<
1118     $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1119     $(POST_PROCESS_O)
1120     $(RM) y.tab.c
1121 .l:
1122     $(RM) $*.c
1123     $(LEX.l) $< > $*.c
1124     $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1125     $(POST_PROCESS)
1126     $(RM) $*.c
1127 .l.o:
1128     $(RM) $*.c
1129     $(LEX.l) $< > $*.c
1130     $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1131     $(POST_PROCESS_O)
1132     $(RM) $*.c

1134 .bin.o:
1135     $(COMPILE.b) -o $@ $<
1136     $(POST_PROCESS_O)

1138 .java.class:
1139     $(COMPILE.java) $<

1141 # Bourne and Korn shell script message catalog build rules.
1142 # We extract all gettext strings with sed(1) (being careful to permit
1143 # multiple gettext strings on the same line), weed out the dups, and
1144 # build the catalogue with awk(1).

1146 .sh.po .ksh.po:
1147     $(SED) -n -e ":a"
1148             -e "h"
1149             -e "s/.*gettext *\([^\"]*\).*\|/p"
1150             -e "x"
1151             -e "s/\(.*\)gettext *\([^\"]*\)(.*)\|/12/"
1152             -e "t a"
1153     $< | sort -u | $(AWK) '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1155 #
1156 # Python and Perl executable and message catalog build rules.
1157 #
1158 .SUFFIXES: .pl .pm .py .pyc

1160 .pl:
1161     $(RM) $@;
1162     $(SED) -e "s@TEXT_DOMAIN@\$(TEXT_DOMAIN)\@" $< > $@;
1163     $(CHMOD) +x $@

1165 .py:
1166     $(RM) $@; $(SED) \
1167         -e "1s:^\#@PYTHON@:\#@$(PYSHEBANG):" \
1168         -e "1s:^\#@TOOLS_PYTHON@:\#@$(TOOLS_PYTHON):" \
1169         < $< > $@; $(CHMOD) +x $@

1171 .py.pyc:
1172     $(RM) $@
1173     $(PYTHON) -mpy_compile $<
1174     @[ $(<)c = $@ ] || $(MV) $(<)c $@

1176 .py.po:
1177     $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;

```

```

1179 .pl.po .pm.po:
1180     $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1181     $(RM)   $@ ;
1182     $(SED)  "/^domain/d" < $(<F).po > $@ ;
1183     $(RM)  $(<F).po

1185 #
1186 # When using xgettext, we want messages to go to the default domain,
1187 # rather than the specified one. This special version of the
1188 # COMPILER.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1189 # causing xgettext to put all messages into the default domain.
1190 #
1191 CPPFORPO=$(COMPILER.cpp:\ "$(TEXT_DOMAIN)"=TEXT_DOMAIN)

1193 .c.i:
1194     $(CPPFORPO) $< > $@

1196 .h.i:
1197     $(CPPFORPO) $< > $@

1199 .y.i:
1200     $(YACC) -d $<
1201     $(CPPFORPO) y.tab.c > $@
1202     $(RM) y.tab.c

1204 .l.i:
1205     $(LEX) $<
1206     $(CPPFORPO) lex.yy.c > $@
1207     $(RM) lex.yy.c

1209 .c.po:
1210     $(CPPFORPO) $< > $<.i
1211     $(BUILD.po)

1213 .cc.po:
1214     $(CPPFORPO) $< > $<.i
1215     $(BUILD.po)

1217 .y.po:
1218     $(YACC) -d $<
1219     $(CPPFORPO) y.tab.c > $<.i
1220     $(BUILD.po)
1221     $(RM) y.tab.c

1223 .l.po:
1224     $(LEX) $<
1225     $(CPPFORPO) lex.yy.c > $<.i
1226     $(BUILD.po)
1227     $(RM) lex.yy.c

1229 #
1230 # Rules to perform stylistic checks
1231 #
1232 .SUFFIXES: .x .xml .check .xmlchk

1234 .h.check:
1235     $(DOT_H_CHECK)

1237 .x.check:
1238     $(DOT_X_CHECK)

1240 .xml.xmlchk:
1241     $(MANIFEST_CHECK)

1243 #

```

```

1244 # Include rules to render automated sccs get rules "safe".
1245 #
1246 include $(SRC)/Makefile.noget

```

```

*****
8747 Wed Apr 24 01:32:50 2019
new/usr/src/cmd/cmd-inet/usr.sbin/Makefile
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
25 # Copyright 2019, Joyent, Inc.
25 # Copyright (c) 2018, Joyent, Inc.
26 #

28 SYNCPROG=      syncinit syncloop syncstat

30 PROG=          6to4relay arp if_mpadm \
31                in.comsat in.fingerd in.rarpd in.rexecd in.rlogind \
32                in.rshd in.rwhod in.telnetd in.tftpd ipaddrsel \
33                ndd ndp $(SYNCPROG)

35 MANIFEST=      rarp.xml telnet.xml comsat.xml finger.xml \
36                login.xml shell.xml rexec.xml socket-filter-kssl.xml
37 SVCMETHOD=     svc-sockfilter

39 ROOTFS_PROG=   hostconfig route soconfig
40 SBINLINKS=     hostconfig route

42 RPCSVCPROG=    hostconfig
43 AUDITPROG=     in.rexecd in.rlogind in.rshd in.telnetd
44 PAMPROG=       in.rexecd in.rlogind in.rshd in.telnetd
45 SOCKETPROG=    6to4relay arp hostconfig if_mpadm in.comsat \
46                in.fingerd in.rarpd in.rexecd in.rlogind in.rshd \
47                in.rwhod in.telnetd in.tftpd ipaddrsel ndp route
48 NSLPROG=       6to4relay arp hostconfig in.comsat in.rarpd \
49                in.rexecd in.rlogind in.rshd in.rwhod in.telnetd \
50                in.tftpd ipaddrsel ndp route
51 CMDPROG=       in.telnetd
52 K5PROGS=       in.telnetd in.rlogind in.rshd
53 TSNETPROG=     route
54 DLADMPROG=     6to4relay
55 DEFAULTFILES=  telnetd.dfl

57 PROGSRCS=      $(PROG:%=%.c)
58 TFTPDOBSJS=    in.tftpd.o tftpsubs.o

```

```

59 OTHERSRC=      ../usr.bin/tftpd/tftpsubs.c
60 K5RLOGINOBSJS= in.rlogind.o
61 K5RSHDOBSJS=   in.rshd.o
62 K5TELNETOBSJS= in.telnetd.o
63 SRCS=          $(PROGSRCS) $(OTHERSRC)

65 SUBDIRS=       ifconfig ilbadm in.rdisc in.routed \
66                in.talkd inetadm inetconv ipadm ipmpstat ipqosconf ipsecutils \
67                kssl/kssladm kssl/ksslcfg nwamadm nwamcfg ping routeadm \
68                snoop spptun traceroute wifonfig

70 MSGSUBDIRS=    ifconfig ilbadm in.routed in.talkd \
71                inetadm inetconv ipadm ipmpstat ipqosconf ipsecutils \
72                kssl/ksslcfg nwamadm nwamcfg routeadm spptun snoop wifonfig

74 # As programs get lint-clean, add them here and to the 'lint' target.
75 # Eventually this hack should go away, and all in PROG should be
76 # lint-clean.
77 LINTCLEAN=      6to4relay arp in.rlogind in.rshd in.telnetd in.tftpd \
78                ipaddrsel ndp route \
79                in.rarpd if_mpadm $(SYNCPROG)
80 # Likewise, as subdirs get lint-clean, add them here. Once
81 # they're all clean, replace the dependency of the lint target
82 # with SUBDIRS. Also (sigh) deal with the commented-out build lines
83 # for the lint rule.
84 LINTSUBDIRS=    ilbadm in.rdisc in.routed in.talkd inetadm \
85                inetconv ipmpstat ipqosconf ipsecutils kssl/kssladm \
86                kssl/ksslcfg nwamadm nwamcfg ping routeadm spptun traceroute \
87                wifonfig
88 # And as programs are verified not to attempt to write into constants,
89 # -xstrconst should be used to ensure they stay that way.
90 CONSTCLEAN=

92 include ../../Makefile.cmd
93 ROOTMANIFESTDIR= $(ROOTSVCNETWORK)
94 $(ROOTMANIFEST) := FILEMODE= 444
95 include ../Makefile.cmd-inet

97 ROOTSBINPROG = $(ROOTFS_PROG:%=$(ROOTSBIN)/%)
98 ROOTUSRSBINLINKS = $(SBINLINKS:%=$(ROOTUSRSBIN)/%)

100 COMMONOBSJS=   addr_match.o kcmd.o store_forw_creds.o
101 COMMONSRCS=     $(COMMONOBSJS:%.o=$(CMDINETCOMMONDIR)/%.c)
102 SRCS+=          $(COMMONSRCS)

104 CERRWARN +=    -_gcc=-Wno-implicit-function-declaration
105 CERRWARN +=    -_gcc=-Wno-uninitialized
106 CERRWARN +=    -_gcc=-Wno-unused-variable
107 CERRWARN +=    -_gcc=-Wno-unused-function
108 CERRWARN +=    -_gcc=-Wno-parentheses
109 CERRWARN +=    -_gcc=-Wno-char-subscripts
110 CERRWARN +=    -_gcc=-Wno-extra
111 CERRWARN +=    -_gcc=-Wno-address

113 # "missing type for func"
114 SMATCH=off

116 #
117 # Message catalog
118 #
119 POFILES=       6to4relay.po if_mpadm.po in.comsat.po ipaddrsel.po route.po \
120                soconfig.po
121 POFILE=        usr.sbin.po

123 all:=          TARGET= all
124 install:=      TARGET= install

```

## new/usr/src/cmd/cmd-inet/usr.sbin/Makefile

3

```

125 clean:=          TARGET= clean
126 clobber:=       TARGET= clobber
127 lint:=          TARGET= lint
128 _msg:=           TARGET= _msg

130 CLOBBERFILES += $(ROOTFS_PROG) $(PROG)
131 CLEANFILES += $(COMMONOBS) $(K5RLOGINOBS) $(K5RSHDOBS) $(TFTPDOBS) \
132               $(K5TELNETOBS)

134 CPPFLAGS +=      -DSYSV -DBSD_COMP -I$(CMDINETCOMMONDIR) -I

136 include $(SRC)/lib/gss_mechs/mech_krb5/Makefile.mech_krb5
137 K5LIBS=

139 # Eventually just plain CFLAGS should be += -v, but not until all in
140 # PROGS are lint clean.
141 $(LINTCLEAN)      :=          CFLAGS += $(CCVERBOSE)
142 $(CONSTCLEAN)    :=          CFLAGS += $(XSTRCONST)

144 $(SYNCPROG)      :=          LDLIBS += -ldlpi
145 $(SOCKETPROG)   :=          LDLIBS += -lsocket
146 $(NSLPROG)      :=          LDLIBS += -lnsl
147 $(AUDITPROG)    :=          LDLIBS += -lbsm
148 $(PAMPROG)      :=          LDLIBS += -lpam
149 $(RPCSVCPROG)   :=          LDLIBS += -lrpcsvc
150 $(K5PROGS)      :=          LDFFLAGS += $(KRUNPATH) \
151                          -L$(ROOT)$(KLIBDIR_DO) -L$(ROOT)$(KLIBDIR_GL)
152 $(K5PROGS)      :=          K5LIBS= -lmec_krb5
153 $(K5PROGS)      :=          CPPFLAGS += -I$(SRC)/head \
154                          -I$(SRC)/uts/common/ \
155                          -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
156                          -I$(SRC)/lib/gss_mechs/mech_krb5/include \
157                          -I$(SRC)/lib/pam_modules/krb5
158 LDLIBS +=        $(K5LIBS)
159 $(TSNETPROG)    :=          LDLIBS += -ltsnet
160 $(DLADMPROG)   :=          LDLIBS += -ldladm

162 in.rarpd        :=          LDLIBS += -linetutil -ldlpi
163 if_mpadm        :=          LDLIBS += -linetutil -lipmp
164 if_mpadm.po     :=          XGETFLAGS += -a
165 route          :=          CPPFLAGS += -DNDEBUG
166 ndd            :=          LDLIBS += -ldladm -lipadm
167 $(RELEASE_BUILD)ndd :=      CERRWARN += -_gcc=-Wno-unused
168 in.comsat       :=          LDFFLAGS += $(MAPFILE.NGB:%=-M%)

170 .KEEP_STATE:

172 .PARALLEL:

174 all: $(PROG) $(ROOTFS_PROG) $(SUBDIRS) THIRDPARTYLICENSE.arp

176 #
177 # message catalog
178 #
179 _msg: $(MSGSUBDIRS) $(POFILE)

181 syncutil: $(SYNCPROG)

183 $(POFILE): $(POFILES)
184     $(RM) $@
185     cat $(POFILES) > $@

187 %.o: $(CMDINETCOMMONDIR)/%.c
188     $(COMPILE.c) -o $@ $<

190 in.telnetd: $(K5TELNETOBS)

```

## new/usr/src/cmd/cmd-inet/usr.sbin/Makefile

4

```

191     $(LINK.c) $(K5TELNETOBS) -o $@ $(LDLIBS)
192     $(POST_PROCESS)

194 in.rlogind: $(K5RLOGINOBS) $(COMMONOBS)
195     $(LINK.c) $(K5RLOGINOBS) $(COMMONOBS) -o $@ $(LDLIBS)
196     $(POST_PROCESS)

198 in.rshd: $(K5RSHDOBS) $(COMMONOBS)
199     $(LINK.c) $(K5RSHDOBS) $(COMMONOBS) -o $@ $(LDLIBS)
200     $(POST_PROCESS)

202 in.tftpd: $(TFTPDOBS)
203     $(LINK.c) $(TFTPDOBS) -o $@ $(LDLIBS)
204     $(POST_PROCESS)

206 tftpsubs.o: $(OTHERSRC)
207     $(COMPILE.c) $(OTHERSRC) -o $@
208     $(POST_PROCESS_o)

210 $(ROOTUSRSBINLINKS):
211     -$(RM) $@; $(SYMLINK) ../../sbin/$(@F) $@

213 install: $(PROG) $(ROOTFS_PROG) $(SUBDIRS) .WAIT $(ROOTUSRSBINPROG) \
214           $(ROOTSBINPROG) $(ROOTUSRSBINLINKS) $(ROOTETCDEFAULTFILES) \
215           $(ROOTMANIFEST) $(ROOTSVCMETHOD) THIRDPARTYLICENSE.arp

217 THIRDPARTYLICENSE.arp: arp.c
218     $(SED) -n 'University of California//SUCH DAMAGE/p' arp.c > $@

220 CLOBBERFILES += THIRDPARTYLICENSE.arp

222 #
223 # The reason this rule checks for the existence of the
224 # Makefile is that some of the directories do not exist
225 # in our exportable source builds.
226 #
227 $(SUBDIRS): FRC
228     @if [ -f $@/Makefile ]; then \
229         cd $@; pwd; $(MAKE) $(TARGET); \
230     else \
231         true; \
232     fi

234 FRC:

236 check: $(CHKMANIFEST)

238 clean: $(SUBDIRS) clean_local
239 clean_local:
240     $(RM) $(CLEANFILES)

242 clobber: $(SUBDIRS) clobber_local
243 clobber_local: clean_local
244     $(RM) $(CLOBBERFILES)

246 lint: $(LINTSUBDIRS)
247     $(LINT.c) 6to4relay.c $(LDLIBS) -lsocket -ldladm
248     $(LINT.c) arp.c $(LDLIBS) -lsocket -lnsl
249     @# $(LINT.c) in.rexecd.c $(LDLIBS) -lbsm -lpam
250     $(LINT.c) -erroff=E_NAME_USED_NOT_DEF2 -erroff=E_NAME_DEF_NOT_USED2 \
251         -I$(SRC)/head -I$(SRC)/uts/common/ \
252         -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
253         -I$(SRC)/lib/gss_mechs/mech_krb5/include \
254         -I$(SRC)/lib/pam_modules/krb5 \
255         in.rlogind.c $(COMMONSRCS) $(LDLIBS) -lbsm -lpam -lsocket -lnsl
256     $(LINT.c) -erroff=E_NAME_USED_NOT_DEF2 -erroff=E_NAME_DEF_NOT_USED2 \

```

```
257         -I$(SRC)/head -I$(SRC)/uts/common/ \
258         -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
259         -I$(SRC)/lib/gss_mechs/mech_krb5/include \
260         -I$(SRC)/lib/pam_modules/krb5 \
261         in.rshd.c $(COMMONSRCS) $(LDLIBS) -lbsd -lpam -lsocket -lnsl
262 $(LINT.c) -erroff=E_NAME_USED_NOT_DEF2 \
263         -erroff=E_GLOBAL_COULD_BE_STATIC2 \
264         -I$(SRC)/head -I$(SRC)/uts/common/ \
265         -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
266         -I$(SRC)/lib/gss_mechs/mech_krb5/include \
267         -I$(SRC)/lib/pam_modules/krb5 \
268         in.telnetd.c $(LDLIBS) -lbsd -lpam -lsocket -lnsl
269 $(LINT.c) if_mpadm.c $(LDLIBS) -lsocket -lnsl -lipmp -linetutil
270 $(LINT.c) ipaddrsel.c $(LDLIBS) -lsocket -lnsl
271 $(LINT.c) route.c $(LDLIBS) -lsocket -lnsl -ltsnet
272 $(LINT.c) syncinit.c $(LDLIBS) -ldlpi
273 $(LINT.c) syncloop.c $(LDLIBS) -ldlpi
274 $(LINT.c) syncstat.c $(LDLIBS) -ldlpi
275 $(LINT.c) -erroff=E_NAME_USED_NOT_DEF2 in.rarpd.c $(LDLIBS) \
276         -lsocket -lnsl
277 $(LINT.c) ndp.c $(LDLIBS) \
278         -lsocket -lnsl
279 $(LINT.c) in.tftpd.c ../usr.bin/tftp/tftpsubs.c $(LDLIBS) \
280         -lsocket -lnsl
```

new/usr/src/cmd/fm/fmd/Makefile.fmd

1

```
*****
6222 Wed Apr 24 01:32:50 2019
new/usr/src/cmd/fm/fmd/Makefile.fmd
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright 2015 RackTop Systems.
24 # Copyright 2019, Joyent, Inc.
25 # Copyright (c) 2018, Joyent, Inc.

27 .KEEP_STATE:

29 SRCS += fmd.c \
30     fmd_api.c \
31     fmd_alloc.c \
32     fmd_asru.c \
33     fmd_buf.c \
34     fmd_builtin.c \
35     fmd_case.c \
36     fmd_ckpt.c \
37     fmd_conf.c \
38     fmd_ctl.c \
39     fmd_dispq.c \
40     fmd_dr.c \
41     fmd_error.c \
42     fmd_event.c \
43     fmd_eventq.c \
44     fmd_fmri.c \
45     fmd_idspace.c \
46     fmd_list.c \
47     fmd_log.c \
48     fmd_main.c \
49     fmd_module.c \
50     fmd_nv.c \
51     fmd_proc.c \
52     fmd_protocol.c \
53     fmd_rpc.c \
54     fmd_rpc_admin.c \
55     fmd_rpc_api.c \
56     fmd_rtld.c \
57     fmd_scheme.c \
58     fmd_self.c \
```

new/usr/src/cmd/fm/fmd/Makefile.fmd

2

```
59     fmd_serdc.c \
60     fmd_string.c \
61     fmd_subr.c \
62     fmd_svc_admin.c \
63     fmd_svc_api.c \
64     fmd_sysevent.c \
65     fmd_thread.c \
66     fmd_time.c \
67     fmd_timerq.c \
68     fmd_topo.c \
69     fmd_trace.c \
70     fmd_ustat.c \
71     fmd_xdr_admin.c \
72     fmd_xdr_api.c \
73     fmd_xprt.c

75 PROG = fmd
76 MANIFEST = ../common/$(PROG).xml

78 MAPFILE_DMODO = $(SRC)/cmd/mdb/common/modules/conf/mapfile-extern

81 LOGADMFILE = $(PROG).logadm.conf
82 LOGADMDIR = $(ROOT)/etc/logadm.d
83 LOGADMENT = $(LOGADMDIR)/$(LOGADMFILE)
84 $(LOGADMENT) := FILEMODE = 444

86 ROOTPDIR = $(ROOT)/usr/lib/fm/$(PROG)
87 ROOTVDIR = $(ROOT)/var/fm/$(PROG)
88 ROOTVSUB = $(ROOTVDIR)/ckpt $(ROOTVDIR)/rsrc $(ROOTVDIR)/xprt
89 ROOTPROG = $(ROOTPDIR)/$(PROG)

91 ROOTMANIFESTDIR = $(ROOTSVCSYSTEM)
92 ROOTMANIFEST = $(ROOTMANIFESTDIR)/$(PROG).xml

94 OBJS = $(SRCS:%.c=%.o)
95 LINTFILES = $(SRCS:%.c=%.ln)

97 CLEANFILES += ../common/fmd_rpc_api.h ../common/fmd_rpc_admin.h
98 CLEANFILES += fmd_svc_admin.c fmd_svc_api.c fmd_xdr_admin.c fmd_xdr_api.c
99 CLEANFILES += ../common/fmd_error.c

101 DMODO = fmd.so
102 ROOTDMODO = $(ROOT)/usr/lib/mdb/proc/$(DMODO)
103 DMODO_SRCS = fmd_mdb.c
104 DMODO_OBJS = $(DMODO_SRCS:%.c=%.o)
105 DMODO_LINT = $(DMODO_SRCS:%.c=%.ln)

107 HDRS = fmd_api.h fmd_fmri.h
108 ROOTCDIR = $(ROOT)/etc/fm/$(PROG)
109 ROOTHDRS = $(ROOT)/usr/include/fm
110 ROOTHDRS = $(HDRS:%=$(ROOTHDRS)/%)

112 $(ROOTHDRS) := FILEMODE = 0644
113 $(ROOTVSUB) := DIRMODE = 0755
114 $(ROOTMANIFEST) := FILEMODE = 0444

116 $(NOT_RELEASE_BUILD)CPPFLAGS += -DDEBUG
117 CPPFLAGS += -I. -I../common
118 CPPFLAGS += -D_POSIX_PTHREAD_SEMANTICS -D_REENTRANT
119 CFLAGS += $(CTF_FLAGS) $(CCVERBOSE) $(XSTRCONST)
120 LINTFLAGS += -mu
121 CERRWARN += -_gcc=-Wno-switch
122 CERRWARN += -_gcc=-Wno-parentheses
123 CERRWARN += -_gcc=-Wno-uninitialized
124 CERRWARN += -_gcc=-Wno-clobbered
```

```

125 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused-but-set-variable
126 CERRWARN += -_gcc=-Wno-unused-variable
125 CERRWARN += -_gcc=-Wno-clobbered

128 # not linted
129 SMATCH=off

131 $(PROG) := LDLFLAGS += -R/usr/lib/fm
132 $(PROG) := LDLIBS += -L$(ROOTLIB)/fm -ltopo -ldiagcode -lsysevent -lsmbios \
133     -luuid -lnvpair -lexacct -lnsl -lumem -ldevinfo -lfmd_msg

135 $(DMOD) := CFLAGS += $(CC_PICFLAGS) -G $(XREGSFLAG)
136 $(DMOD) := LDLFLAGS += $(ZTEXT) $(ZDEFS) $(MAPFILE-DMOD:%=-M%)

138 #
139 # rpcgen(1) produces code that wants to be in the foreground if we're compiled
140 # DEBUG, which isn't appropriate for our daemon. Forcibly undefine this code.
141 # It also produces various lint warnings; turn these off for these files only.
142 # Unfortunately we also have to turn off IGNOR2 for pass2 lint as well.
143 #
144 fmd_svc_adm.o fmd_svc_api.o := CPPFLAGS += -UDEBUG -URPC_SVC_FG
145 fmd_xdr_adm.o fmd_xdr_api.o := CPPFLAGS += -UDEBUG -URPC_SVC_FG

147 fmd_svc_adm.ln fmd_svc_api.ln fmd_xdr_adm.ln fmd_xdr_api.ln := LINTFLAGS += \
148     -xerroff=E_FUNC_ARG_UNUSED -xerroff=E_FUNC_VAR_UNUSED -xerroff=E_STATIC_UNUSED

150 lint_prog := LINTFLAGS += -xerroff=E_FUNC_RET_ALWAYS_IGNORE2

152 .NO_PARALLEL:
153 .PARALLEL: $(OBJS) $(LINTFILES)

155 all: $(PROG) $(DMOD) install_h

157 $(LOGADMDIR):
158     $(INS.dir)

160 $(LOGADMDIR)/%.conf: ../common/%.conf
161     $(INS.file)

163 $(PROG): $(OBJS)
164     $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
165     $(CTFMERGE) -L VERSION -o $@ $(OBJS)
166     $(POST_PROCESS)

168 $(DMOD): $(DMOD_OBJS)
169     $(LINK.c) $(DMOD_OBJS) -o $@ $(LDLIBS) -lc
170     $(POST_PROCESS)

172 %.o: ../common/%.c
173     $(COMPILE.c) $<
174     $(CTFCONVERT_O)

176 %.o: %.c
177     $(COMPILE.c) $<
178     $(CTFCONVERT_O)

180 ../common/fmd_error.c: ../common/mkerror.sh ../common/fmd_error.h
181     sh ../common/mkerror.sh < ../common/fmd_error.h > $@

183 ../common/fmd_rpc_%.c: ../common/fmd_rpc_%.h

185 ../common/fmd_rpc.c: ../common/fmd_rpc_adm.h ../common/fmd_rpc_api.h

187 ../common/fmd_rpc_%.h: ../common/fmd_rpc_%.x
188     $(RPCGEN) -CMN -h -o $@ $<

```

```

190 fmd_svc_%.c: ../common/fmd_rpc_%.h
191     $(RPCGEN) -CMN -m -o $@ `echo $< | sed 's/h$$/x/'`

193 fmd_xdr_%.c: ../common/fmd_rpc_%.h
194     $(RPCGEN) -CMN -c -o $@ `echo $< | sed 's/h$$/x/'`

196 clean:
197     $(RM) $(OBJS) $(DMOD_OBJS) $(LINTFILES) $(DMOD_LINT)
198     $(RM) $(CLEANFILES)

200 clobber: clean
201     $(RM) $(PROG) $(DMOD)

203 %.ln: ../common/%.c
204     $(LINT.c) -c $<

206 %.ln: %.c
207     $(LINT.c) -c $<

209 lint_prog: $(LINTFILES)
210     $(LINT) $(LINTFLAGS) $(LINTFILES) $(LDLIBS)

212 lint_dmod: $(DMOD_LINT)
213     $(LINT) $(LINTFLAGS) $(DMOD_LINT) $(LDLIBS)

215 lint: lint_prog lint_dmod

217 $(ROOT)/etc/fm:
218     $(INS.dir)

220 $(ROOTCDIR): $(ROOT)/etc/fm
221     $(INS.dir)

223 $(ROOTHDIR):
224     $(INS.dir)

226 $(ROOTHDIR)/%.h: ../common/%.h
227     $(INS.file)

229 $(ROOTMANIFESTDIR)/%.xml: ../common/%.xml
230     $(INS.file)

232 $(ROOT)/var/%:
233     $(INS.dir)

235 $(ROOTVDIR): $(ROOT)/var/fm
236     $(INS.dir)

238 $(ROOTVSUB): $(ROOTVDIR)
239     $(INS.dir)

241 $(ROOT)/usr/lib/fm:
242     $(INS.dir)

244 $(ROOTPDIR): $(ROOT)/usr/lib/fm
245     $(INS.dir)

247 $(ROOTPROG): $(ROOTPDIR) $(PROG)
248     $(RM) $@; $(INS) -s -m 0555 -f $(@) $(PROG)

250 $(ROOT)/usr/lib/mdb/proc: $(ROOT)/usr/lib/mdb
251     $(INS.dir)

253 $(ROOTDMOD): $(ROOT)/usr/lib/mdb/proc $(DMOD)
254     $(RM) $@; $(INS) -s -m 0555 -f $(@) $(DMOD)

```

**new/usr/src/cmd/fm/fmd/Makefile.fmd**

5

256 install\_h: \$(ROOTHDR) \$(ROOTHDRS)

258 install: all install\_h \$(ROOTPROG) \$(ROOTDMOD) \$(LOGADMDIR) \$(LOGADMENT) \  
259 \$(ROOTCDIR) \$(ROOTVDIR) \$(ROOTVSUB) \$(ROOTMANIFEST)

261 check: \$(CHKMANIFEST)



new/usr/src/cmd/mdb/intel/Makefile.kmdb

1

```
*****
1900 Wed Apr 24 01:32:50 2019
new/usr/src/cmd/mdb/intel/Makefile.kmdb
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Copyright 2019, Joyent, Inc.
25 # Copyright 2018 Joyent, Inc.
26 #

28 PROMSRCS += \
29     prom_env.c \
30     prom_getchar.c \
31     prom_init.c \
32     prom_printf.c \
33     prom_putchar.c

35 KMDBSRCS += \
36     kaif.c \
37     kmdb_dpi_isadep.c \
38     kmdb_fault_isadep.c \
39     kmdb_kdi_isadep.c \
40     kmdb_promif_isadep.c \
41     kvm_isadep.c

43 KMDBML += \
44     kmdb_asmutil.s \
45     kmdb_setcontext.s

47 KCTLSRCS += \
48     kctl_isadep.c

50 CTXOFFUSERS = \
51     kmdb_setcontext.o

53 CPPFLAGS += -DDIS_TEXT

55 $(CTXOFFUSERS) $(CTXOFFUSERS:%.o=%.ln): kmdb_context_off.h

57 kaif_activate.o kaif_activate.ln := CPPFLAGS += -D_MACHDEP -D_KMEMUSER
```

new/usr/src/cmd/mdb/intel/Makefile.kmdb

2

```
59 $(RELEASE_BUILD)kaif.o := CERRWARN += -_gcc=-Wno-unused

61 STANLIBS += \
62     ../libstandctf/libstandctf.so \
63     $(SRC)/lib/libumem/$(MACHDIR)/libstandumem.so \
64     ../libstand/libstand.a

66 KMDBLIBS = $(STANLIBS) ../mdb_ks/kmod/mdb_ks

68 MAPFILE_SOURCES = \
69     $(MAPFILE_SOURCES_COMMON) \
70     ../../kmdb/kmdb_dpi_isadep.h \
71     $(MAPFILE_SOURCES_$(MACH))

73 %.o: ../../../../uts/intel/promif/%.c
74     $(COMPILE.c) $<
75     $(CTFCONVERT_O)

77 %.ln: ../../../../uts/intel/promif/%.c
78     $(LINT.c) -c $<
```

```

*****
4822 Wed Apr 24 01:32:51 2019
new/usr/src/cmd/sgs/libld/Makefile.com
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # Copyright (c) 2018, Joyent, Inc.
27 # Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
28 #
29 LIBRARY = libld.a
30 VERS = .4
31 #
32 COMOBS = debug.o globals.o util.o
33 #
34 COMOBS32 = args32.o entry32.o exit32.o groups32.o \
35 ldentry32.o ldlibs32.o ldmachdep32.o ldmain32.o \
36 libs32.o files32.o map32.o map_core32.o \
37 map_support32.o map_v232.o order32.o outfile32.o \
38 place32.o relocate32.o resolve32.o sections32.o \
39 sunwmove32.o support32.o syms32.o update32.o \
40 unwind32.o version32.o wrap32.o
41 #
42 COMOBS64 = args64.o entry64.o exit64.o groups64.o \
43 ldentry64.o ldlibs64.o ldmachdep64.o ldmain64.o \
44 libs64.o files64.o map64.o map_core64.o \
45 map_support64.o map_v264.o order64.o outfile64.o \
46 place64.o relocate64.o resolve64.o sections64.o \
47 sunwmove64.o support64.o syms64.o update64.o \
48 unwind64.o version64.o wrap64.o
49 #
50 TOOLOBS = alist.o assfail.o findprime.o string_table.o \
51 strhash.o
52 AVLOBJ = avl.o
53 #
54 # Relocation engine objects.
55 G_MACHOBS32 = doreloc_sparc_32.o doreloc_x86_32.o
56 G_MACHOBS64 = doreloc_sparc_64.o doreloc_x86_64.o
57 #
58 # Target specific objects (sparc/sparcv9)
59 L_SPARC_MACHOBS32 = machrel.sparc32.o machsym.sparc32.o

```

```

60 L_SPARC_MACHOBS64 = machrel.sparc64.o machsym.sparc64.o
61 #
62 # Target specific objects (i386/amd64)
63 E_X86_TOOLOBS = lebl28.o
64 L_X86_MACHOBS32 = machrel.intel32.o
65 L_X86_MACHOBS64 = machrel.amd64.o
66 #
67 # All target specific objects rolled together
68 E_TOOLOBS = $(E_SPARC_TOOLOBS) \
69 $(E_X86_TOOLOBS)
70 L_MACHOBS32 = $(L_SPARC_MACHOBS32) \
71 $(L_X86_MACHOBS32)
72 L_MACHOBS64 = $(L_SPARC_MACHOBS64) \
73 $(L_X86_MACHOBS64)
74 #
75 #
76 BLTOBJ = msg.o
77 ELFCAPOBJ = elfcap.o
78 #
79 OBJECTS = $(BLTOBJ) $(G_MACHOBS32) $(G_MACHOBS64) \
80 $(L_MACHOBS32) $(L_MACHOBS64) \
81 $(COMOBS) $(COMOBS32) $(COMOBS64) \
82 $(TOOLOBS) $(E_TOOLOBS) $(AVLOBJ) $(ELFCAPOBJ)
83 #
84 include $(SRC)/lib/Makefile.lib
85 include $(SRC)/cmd/sgs/Makefile.com
86 #
87 SRCDIR = ../common
88 #
89 CERRWARN += _gcc=-Wno-unused-value
90 CERRWARN += _gcc=-Wno-parentheses
91 CERRWARN += _gcc=-Wno-uninitialized
92 CERRWARN += _gcc=-Wno-switch
93 CERRWARN += _gcc=-Wno-char-subscripts
94 CERRWARN += _gcc=-Wno-type-limits
95 $(RELEASE_BUILD)CERRWARN += _gcc=-Wno-unused
96 #
97 SMOFF += no_if_block
98 #
99 # Location of the shared relocation engines maintained under usr/src/uts.
100 #
101 KRTLDIR = $(SRCBASE)/uts/$(VAR_PLAT_i386)/krtld
102 KRTLDIR_AMD64 = $(SRCBASE)/uts/$(VAR_PLAT_amd64)/krtld
103 KRTLDIR_SPARC = $(SRCBASE)/uts/$(VAR_PLAT_sparc)/krtld
104 #
105 #
106 CPPFLAGS += -DUSE_LIBLD_MALLOC -I$(SRCBASE)/lib/libc/inc \
107 -I$(SRCBASE)/uts/common/krtld -I$(SRCBASE)/uts/sparc \
108 $(VAR_LIBLD_CPPFLAGS)
109 LDLIBS += $(CONVLIBDIR) $(CONV_LIB) $(LDBGLIBDIR) $(LDBGLIB) \
110 $(ELFLIBDIR) -lself $(DLLIB) -lc
111 #
112 DYNFLG += $(VERSREF) $(CC_USE_PROTO) '-R$ORIGIN'
113 #
114 native:= DYNFLG += $(CONVLIBDIR)
115 #
116 BLTDEFS = msg.h
117 BLTDATA = msg.c
118 BLTMESG = $(SGMSGDIR)/libld
119 #
120 BLTFILES = $(BLTDEFS) $(BLTDATA) $(BLTMESG)
121 #
122 # Due to cross linking support, every copy of libld contains every message.
123 # However, we keep target specific messages in their own separate files for
124 # organizational reasons.
125 #

```

```
126 SGMSGCOM = ../common/libld.msg
127 SGMSGSPARC = ../common/libld.sparc.msg
128 SGMSGINTEL = ../common/libld.intel.msg
129 SGMSGTARG = $(SGMSGCOM) $(SGMSGSPARC) $(SGMSGINTEL)
130 SGMSGALL = $(SGMSGCOM) $(SGMSGSPARC) $(SGMSGINTEL)

132 SGMSGFLAGS1 = $(SGMSGFLAGS) -m $(BLTMSG)
133 SGMSGFLAGS2 = $(SGMSGFLAGS) -h $(BLTDEFS) -d $(BLTDATA) -n libld_msg

135 CHKSRC = $(SRCBASE)/uts/common/krtld/reloc.h \
136 $(COMOBS32:%32.o=../common/%.c) \
137 $(L_MACHOBS32:%32.o=../common/%.c) \
138 $(L_MACHOBS64:%64.o=../common/%.c) \
139 $(KRTLD_I386)/doreloc.c \
140 $(KRTLD_AMD64)/doreloc.c \
141 $(KRTLD_SPARC)/doreloc.c

143 LIBSRC = $(TOOLOBS:%.o=$(SGSTOOLS)/common/%.c) \
144 $(E_TOOLOBS:%.o=$(SGSTOOLS)/common/%.c) \
145 $(COMOBS:%.o=../common/%.c) \
146 $(AVLOBS:%.o=$(VAR_AVLDIR)/%.c) \
147 $(BLTDATA)

149 CLEANFILES += $(BLTFILES)
150 CLOBBERFILES += $(DYNLIB) $(LIBLINKS)

152 ROOTFS_DYNLIB = $(DYNLIB:%=$(ROOTFS_LIBDIR)/%)
```

```

*****
22116 Wed Apr 24 01:32:51 2019
new/usr/src/lib/libc/amd64/Makefile
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2016 Joyent, Inc.
25 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
26 # Copyright 2013 Garrett D'Amore <garrett@damore.org>
27 # Copyright 2018 Nexenta Systems, Inc.
28 # Copyright (c) 2019, Joyent, Inc.
29 # Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
30 #
31 #
32 LIBCBASE= .
33 LIBCDIR=  ${SRC}/lib/libc
34 LIBRARY=  libc.a
35 LIB_PIC=  libc_pic.a
36 VERS=    .1
37 CPP=     /usr/lib/cpp
38 TARGET_ARCH= amd64
39 #
40 # include comm page definitions
41 include ${SRC}/lib/commpage/Makefile.shared.com
42 include ${SRC}/lib/commpage/Makefile.shared.targ
43 #
44 # objects are grouped by source directory
45 #
46 # local objects
47 STRETS=
48 #
49 CRTOBSJ= \
50   cerror.o
51 #
52 DYNOBJS=
53 #
54 FPOBJS= \
55   fpgetmask.o \
56   fpgetround.o \
57   fpsetmask.o \
58   fpsetround.o \
59   fpstart.o \

```

```

60   ieee.o
61 #
62 I386FPOBJS= \
63   _D_cplx_div.o \
64   _D_cplx_div_ix.o \
65   _D_cplx_div_rx.o \
66   _D_cplx_lr_div.o \
67   _D_cplx_lr_div_ix.o \
68   _D_cplx_lr_div_rx.o \
69   _D_cplx_mul.o \
70   _F_cplx_div.o \
71   _F_cplx_div_ix.o \
72   _F_cplx_div_rx.o \
73   _F_cplx_lr_div.o \
74   _F_cplx_lr_div_ix.o \
75   _F_cplx_lr_div_rx.o \
76   _F_cplx_mul.o \
77   _X_cplx_div.o \
78   _X_cplx_div_ix.o \
79   _X_cplx_div_rx.o \
80   _X_cplx_lr_div.o \
81   _X_cplx_lr_div_ix.o \
82   _X_cplx_lr_div_rx.o \
83   _X_cplx_mul.o
84 #
85 FPASMOBJS= \
86   __xgetRD.o \
87   _xtoll.o \
88   _xtoull.o \
89   _base_il.o \
90   fpcw.o \
91   fpgetsticky.o \
92   fpsetsticky.o
93 #
94 ATOMICOBJS= \
95   atomic.o
96 #
97 CHACHAOBJS= \
98   chacha.o
99 #
100 XATTOBJS= \
101   xattr_common.o
102 COMOBJS= \
103   bcmp.o \
104   bcopy.o \
105   bsearch.o \
106   bzero.o \
107   qsort.o \
108   strtol.o \
109   strtoul.o \
110   strtoll.o \
111   strtoull.o
112 #
113 GENOBJS= \
114   $(COMMPAGE_OBJS) \
115   _getsp.o \
116   abs.o \
117   alloca.o \
118   arc4random.o \
119   arc4random_uniform.o \
120   attrat.o \
121   byteorder.o \
122   cuexit.o \
123   ecvt.o \
124   endian.o \
125   errlst.o \

```

## new/usr/src/lib/libc/amd64/Makefile

```

126     amd64_data.o  \
127     ldivide.o    \
128     lock.o       \
129     makecxtxt.o \
130     memccpy.o    \
131     memchr.o     \
132     memcmp.o     \
133     memcpy.o     \
134     memset.o     \
135     new_list.o   \
136     proc64_id.o  \
137     proc64_support.o \
138     setjmp.o     \
139     siginfolst.o \
140     siglongjmp.o \
141     strcmp.o     \
142     strcpy.o     \
143     strlen.o     \
144     strncmp.o    \
145     strncpy.o    \
146     strnlen.o   \
147     sync_instruction_memory.o

```

```

149 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
150 # This macro should ALWAYS be empty; native APIs are already 'large file'.
151 COMSYSOBS64=

```

```
153 SYSOBS64=
```

```

155 COMSYSOBS= \
156     __clock_timer.o \
157     __getloadavg.o  \
158     __rusagesys.o  \
159     __signotify.o   \
160     __sigrt.o       \
161     __time.o        \
162     _lgrp_home_fast.o \
163     _lgrpsys.o     \
164     _nfssys.o      \
165     _portfs.o      \
166     _pset.o        \
167     _rpcsys.o      \
168     _sigaction.o   \
169     _so_accept.o   \
170     _so_bind.o     \
171     _so_connect.o  \
172     _so_getpeername.o \
173     _so_getsockname.o \
174     _so_getsockopt.o \
175     _so_listen.o   \
176     _so_recv.o     \
177     _so_recvfrom.o \
178     _so_recvmsg.o  \
179     _so_send.o     \
180     _so_sendmsg.o  \
181     _so_sendto.o   \
182     _so_setsockopt.o \
183     _so_shutdown.o \
184     _so_socket.o   \
185     _so_socketpair.o \
186     _sockconfig.o  \
187     acct.o         \
188     acl.o          \
189     adjtime.o     \
190     alarm.o       \
191     brk.o

```

3

## new/usr/src/lib/libc/amd64/Makefile

```

192     chdir.o       \
193     chroot.o     \
194     cladm.o      \
195     close.o      \
196     execve.o     \
197     exit.o       \
198     facr.o       \
199     fchdir.o     \
200     fchroot.o   \
201     fdsync.o     \
202     fpathconf.o \
203     fstatfs.o   \
204     fstatvfs.o  \
205     getcpuid.o  \
206     getdents.o  \
207     getegid.o   \
208     geteuid.o   \
209     getgid.o    \
210     getgroups.o \
211     gethrtime.o \
212     getitimer.o \
213     getmsg.o     \
214     getpid.o    \
215     getpmsg.o   \
216     getppid.o   \
217     getrandom.o \
218     getrlimit.o \
219     getuid.o    \
220     gtty.o      \
221     install_utrap.o \
222     ioctl.o     \
223     kaio.o      \
224     kill.o      \
225     llseek.o   \
226     lseek.o    \
227     mmapobjsys.o \
228     memcntl.o   \
229     mincore.o  \
230     mmap.o     \
231     modctl.o   \
232     mount.o    \
233     mprotect.o \
234     munmap.o   \
235     nice.o     \
236     ntp_adjtime.o \
237     ntp_gettime.o \
238     p_online.o \
239     pathconf.o \
240     pause.o    \
241     pcsample.o \
242     pipe2.o    \
243     pollsys.o \
244     pread.o    \
245     preadv.o   \
246     priocntlset.o \
247     processor_bind.o \
248     processor_info.o \
249     profil.o   \
250     psecflagsset.o \
251     putmsg.o   \
252     putpmsg.o  \
253     pwrite.o   \
254     pwritev.o  \
255     read.o     \
256     readv.o    \
257     resolvepath.o

```

4

```

258     seteguid.o      \
259     setgid.o        \
260     setgroups.o     \
261     setitimer.o     \
262     setreid.o       \
263     setrlimit.o    \
264     setuid.o        \
265     sigaltstk.o     \
266     sigprocmsk.o   \
267     sigsendset.o   \
268     sigsuspend.o   \
269     statfs.o        \
270     statvfs.o       \
271     stty.o          \
272     sync.o          \
273     sysconfig.o    \
274     sysfs.o         \
275     sysinfo.o       \
276     syslpw.o        \
277     times.o         \
278     ulimit.o        \
279     umask.o         \
280     umount2.o      \
281     utssys.o        \
282     uucopy.o        \
283     vhangup.o       \
284     waitid.o        \
285     write.o         \
286     writev.o       \
287     yield.o        \
\
289 SYSOBJS= \
290     __clock_gettime.o \
291     __clock_gettime_sys.o \
292     __getcontext.o \
293     __uadmin.o \
294     __lwp_mutex_unlock.o \
295     __stack_grow.o \
296     door.o \
297     forkx.o \
298     forkallx.o \
299     getcontext.o \
300     gettimeofday.o \
301     lwp_private.o \
302     nuname.o \
303     syscall.o \
304     sysi86.o \
305     tls_get_addr.o \
306     uadmin.o \
307     umount.o \
308     uname.o \
309     vforkx.o \
\
311 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
312 # This macro should ALWAYS be empty; native APIs are already 'large file'.
313 PORTGEN64=
\
315 # objects from source under $(LIBCDIR)/port
316 PORTFP= \
317     __flt_decim.o \
318     __flt_rounds.o \
319     __tbl_10_b.o \
320     __tbl_10_h.o \
321     __tbl_10_s.o \
322     __tbl_2_b.o \
323     __tbl_2_h.o \

```

```

324     __tbl_2_s.o \
325     __tbl_fdq.o \
326     __tbl_tens.o \
327     __x_power.o \
328     _base_sup.o \
329     aconvert.o \
330     decimal_bin.o \
331     double_decim.o \
332     econvert.o \
333     fconvert.o \
334     file_decim.o \
335     finite.o \
336     fp_data.o \
337     func_decim.o \
338     gconvert.o \
339     hex_bin.o \
340     ieee_globals.o \
341     pack_float.o \
342     sigfpe.o \
343     string_decim.o \
\
345 PORTGEN= \
346     _env_data.o \
347     _xftw.o \
348     a64l.o \
349     abort.o \
350     addsev.o \
351     ascii_strcasecmp.o \
352     ascii_strncasecmp.o \
353     assert.o \
354     atexit.o \
355     atfork.o \
356     atof.o \
357     atoi.o \
358     atol.o \
359     atoll.o \
360     attropen.o \
361     basename.o \
362     calloc.o \
363     catgets.o \
364     catopen.o \
365     cfgetispeed.o \
366     cfgetospeed.o \
367     cfree.o \
368     cfsetispeed.o \
369     cfsetospeed.o \
370     cftime.o \
371     clock.o \
372     closedir.o \
373     closefrom.o \
374     confstr.o \
375     crypt.o \
376     csetlen.o \
377     ctime.o \
378     ctime_r.o \
379     daemon.o \
380     deflt.o \
381     directio.o \
382     dirname.o \
383     div.o \
384     drand48.o \
385     dup.o \
386     env_data.o \
387     err.o \
388     errno.o \
389     euclen.o \

```

```

390     event_port.o      \|
391     execvp.o          \|
392     explicit_bzero.o \|
393     fattach.o         \|
394     fdetach.o         \|
395     fdopendir.o       \|
396     ffs.o             \|
397     flock.o           \|
398     fls.o             \|
399     fmsg.o            \|
400     freezero.o        \|
401     ftime.o           \|
402     ftok.o            \|
403     fts.o             \|
404     ftw.o             \|
405     gcvt.o            \|
406     get_nprocs.o     \|
407     getauxv.o         \|
408     getcwd.o          \|
409     getdate_err.o    \|
410     getdtablesize.o \|
411     getentropy.o     \|
412     getenv.o         \|
413     getexecname.o    \|
414     getgrnam.o       \|
415     getgrnam_r.o     \|
416     gethostid.o      \|
417     gethostname.o    \|
418     gethz.o           \|
419     getisax.o         \|
420     getloadavg.o     \|
421     getlogin.o       \|
422     getmntent.o      \|
423     getnetgrent.o    \|
424     getopt.o         \|
425     getopt_long.o   \|
426     getpagesize.o   \|
427     getpw.o          \|
428     getpwnam.o       \|
429     getpwnam_r.o     \|
430     getrusage.o      \|
431     getspent.o       \|
432     getspent_r.o     \|
433     getsubopt.o      \|
434     gettxt.o         \|
435     getusershell.o  \|
436     getut.o          \|
437     getutx.o         \|
438     getvfsent.o      \|
439     getwd.o          \|
440     getwidth.o       \|
441     getxby_door.o   \|
442     gtxt.o           \|
443     hsearch.o        \|
444     iconv.o          \|
445     imaxabs.o        \|
446     imaxdiv.o        \|
447     index.o          \|
448     initgroups.o    \|
449     insque.o         \|
450     isaexec.o        \|
451     isastream.o      \|
452     isatty.o         \|
453     killpg.o         \|
454     klpdlib.o        \|
455     l64a.o           \|

```

```

456     lckpwdf.o        \|
457     lconstants.o    \|
458     lexpl0.o         \|
459     lfind.o          \|
460     lfmt.o           \|
461     lfmt_log.o       \|
462     lldiv.o          \|
463     llog10.o         \|
464     lltostr.o        \|
465     lmath.o          \|
466     localtime.o     \|
467     lsearch.o        \|
468     madvise.o        \|
469     malloc.o         \|
470     memalign.o       \|
471     memmem.o         \|
472     memset_s.o       \|
473     mkdev.o          \|
474     mkdtemp.o        \|
475     mkfifo.o         \|
476     mkstemp.o        \|
477     mktemp.o         \|
478     mlock.o          \|
479     mlockall.o       \|
480     mon.o            \|
481     msync.o          \|
482     munlock.o        \|
483     munlockall.o    \|
484     ndbm.o           \|
485     nftw.o           \|
486     nlspath_checks.o \|
487     nsparse.o        \|
488     nss_common.o     \|
489     nss_dbdefs.o     \|
490     nss_deffinder.o  \|
491     opendir.o        \|
492     opt_data.o       \|
493     perror.o         \|
494     pfmt.o           \|
495     pfmt_data.o      \|
496     pfmt_print.o    \|
497     pipe.o           \|
498     plock.o          \|
499     poll.o           \|
500     posix_fadvise.o  \|
501     posix_fallocate.o \|
502     posix_madvise.o  \|
503     posix_memalign.o \|
504     priocntl.o       \|
505     priv_str_xlate.o \|
506     privlib.o        \|
507     psecflags.o      \|
508     psiginfo.o       \|
509     psignal.o        \|
510     pt.o             \|
511     putpwent.o       \|
512     putspent.o       \|
513     raise.o          \|
514     rand.o           \|
515     random.o         \|
516     rctlops.o        \|
517     readdir.o        \|
518     readdir_r.o      \|
519     reallocarray.o   \|
520     reallocarray.o   \|
521     realpath.o       \|

```

```

522     reboot.o           \
523     regexpr.o          \
524     remove.o           \
525     rewinddir.o        \
526     rindex.o           \
527     scandir.o          \
528     seekdir.o          \
529     select.o           \
530     set_constraint_handler_s.o \
531     setlabel.o         \
532     setpriority.o      \
533     settimeofday.o    \
534     sh_locks.o         \
535     sigflag.o          \
536     siglist.o          \
537     sigsend.o          \
538     sigsetops.o        \
539     signal.o           \
540     stack.o            \
541     stpcpy.o           \
542     stpncpy.o          \
543     str2sig.o          \
544     strcase_charmap.o \
545     strcat.o           \
546     strchr.o           \
547     strchrnul.o        \
548     strcspn.o          \
549     strdup.o           \
550     strerror.o         \
551     strlcat.o          \
552     strlcpy.o          \
553     strncat.o          \
554     strndup.o          \
555     strpbk.o           \
556     strrchr.o          \
557     strsep.o           \
558     strsignal.o        \
559     strspn.o           \
560     strstr.o           \
561     strtod.o           \
562     strtointmax.o      \
563     strtok.o           \
564     strtok_r.o         \
565     strtonum.o         \
566     strtoumax.o        \
567     swab.o             \
568     swapctl.o          \
569     sysconf.o          \
570     syslog.o           \
571     tcdrain.o          \
572     tcflow.o           \
573     tcflush.o          \
574     tcgetattr.o        \
575     tcgetpgrp.o        \
576     tcgetsid.o         \
577     tcsendbreak.o     \
578     tcsetattr.o        \
579     tcsetpgrp.o        \
580     tell.o             \
581     telldir.o          \
582     tfind.o            \
583     time_data.o        \
584     time_gdata.o       \
585     timespec_get.o     \
586     tls_data.o         \
587     truncate.o         \

```

```

588     tsdalloc.o        \
589     tsearch.o         \
590     ttyname.o         \
591     ttyslot.o         \
592     ualarm.o          \
593     ucred.o           \
594     valloc.o          \
595     vlfmt.o           \
596     vpfmt.o           \
597     waitpid.o         \
598     walkstack.o       \
599     wdata.o           \
600     xgetwidth.o       \
601     xpg4.o            \
602     xpg6.o            \
\
604 PORTINET=           \
605     inet_lnaof.o      \
606     inet_makeaddr.o  \
607     inet_network.o   \
608     inet_ntoa.o      \
609     inet_ntop.o      \
610     inet_pton.o      \
\
612 PORTPRINT_W=        \
613     doprnt_w.o       \
\
615 PORTPRINT=          \
616     asprintf.o       \
617     doprnt.o         \
618     fprintf.o        \
619     printf.o         \
620     sprintf.o        \
621     vsprintf.o       \
622     vfprintf.o       \
623     vprintf.o        \
624     vsnprintf.o     \
625     vsprintf.o       \
626     vwprintf.o       \
627     wprintf.o        \
\
629 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
630 # This macro should ALWAYS be empty; native APIs are already 'large file'.
631 PORTSTDIO64=
\
633 PORTSTDIO_W=        \
634     doscan_w.o       \
\
636 PORTSTDIO=          \
637     __extensions.o   \
638     __endopen.o      \
639     __filbuf.o        \
640     __findbuf.o      \
641     __flsbuf.o       \
642     __wrtchk.o       \
643     clearerr.o        \
644     ctermid.o        \
645     ctermid_r.o      \
646     cuserid.o         \
647     data.o            \
648     doscan.o          \
649     fdopen.o          \
650     feof.o            \
651     ferror.o          \
652     fgetc.o           \
653     fgets.o           \

```



```

654      fileno.o          \|
655      flockf.o         \|
656      flush.o          \|
657      fopen.o          \|
658      fpos.o           \|
659      fputc.o          \|
660      fputs.o          \|
661      fread.o          \|
662      fseek.o          \|
663      fseeko.o         \|
664      ftell.o          \|
665      ftello.o         \|
666      fwrite.o         \|
667      getc.o           \|
668      getchar.o        \|
669      getline.o        \|
670      getpass.o        \|
671      gets.o           \|
672      getw.o           \|
673      mse.o            \|
674      popen.o          \|
675      putc.o           \|
676      putchar.o       \|
677      puts.o           \|
678      putw.o           \|
679      rewind.o         \|
680      scanf.o          \|
681      setbuf.o         \|
682      setbuffer.o     \|
683      setvbuf.o        \|
684      system.o         \|
685      tempnam.o        \|
686      tmpfile.o        \|
687      tmpnam_r.o       \|
688      ungetc.o         \|
689      vscanf.o         \|
690      vwscanf.o        \|
691      wscanf.o         \|

693 PORTI18N=          \|
694      getwchar.o       \|
695      putwchar.o       \|
696      putws.o          \|
697      strtows.o        \|
698      wcsnlen.o        \|
699      wcsstr.o         \|
700      wcstoimax.o      \|
701      wcstol.o         \|
702      wcstoul.o        \|
703      wcswcs.o         \|
704      wmemchr.o        \|
705      wmemcmp.o        \|
706      wmemcpy.o        \|
707      wmemmove.o       \|
708      wmemset.o        \|
709      wscat.o          \|
710      wschr.o          \|
711      wscmp.o          \|
712      wscopy.o         \|
713      wcsnlen.o        \|
714      wsdup.o          \|
715      wslen.o          \|
716      wsnecat.o        \|
717      wsnncpy.o        \|
718      wsnpcpy.o        \|
719      wspbrk.o         \|

```

```

720      wsprintf.o       \|
721      wsrchr.o         \|
722      wsscanf.o        \|
723      wsspn.o          \|
724      wstod.o          \|
725      wstok.o          \|
726      wstol.o          \|
727      wstoll.o         \|
728      wxfrm.o          \|
729      gettext.o        \|
730      gettext_gnu.o    \|
731      gettext_real.o   \|
732      gettext_util.o   \|
733      plural_parser.o  \|
734      wdresolve.o      \|
735      _ctype.o         \|
736      isascii.o        \|
737      toascii.o        \|

739 PORTI18N_COND=     \|
740      wcstol_longlong.o \|
741      wcstoul_longlong.o \|

743 PORTLOCALE=        \|
744      big5.o           \|
745      btowc.o          \|
746      collate.o        \|
747      collcmp.o        \|
748      euc.o            \|
749      fnmatch.o        \|
750      fgetwc.o         \|
751      fgetws.o         \|
752      fix_grouping.o   \|
753      fputwc.o         \|
754      fputws.o         \|
755      fwide.o          \|
756      gb18030.o        \|
757      gb2312.o         \|
758      gbk.o            \|
759      getdate.o        \|
760      isdigit.o        \|
761      iswctype.o       \|
762      ldpert.o         \|
763      lmessages.o      \|
764      lnumeric.o       \|
765      lmonetary.o      \|
766      localeconv.o     \|
767      localeimpl.o     \|
768      mbftowc.o        \|
769      mblen.o          \|
770      mbrlen.o         \|
771      mbrtowc.o        \|
772      mbsinit.o        \|
773      mbsnrto wcs.o    \|
774      mbsrtowcs.o      \|
775      mbstowcs.o       \|
776      mbtowc.o         \|
777      mskanji.o        \|
778      nextwctype.o     \|
779      nl_langinfo.o    \|
780      none.o           \|
781      rune.o           \|
782      runetype.o       \|
783      setlocale.o      \|
784      setrunelocale.o  \|
785      strcasecmp.o     \|

```

```

786      strcasestr.o      \
787      strcoll.o        \
788      strfmon.o        \
789      strftime.o       \
790      strncasecmp.o    \
791      strtptime.o     \
792      strxfrm.o        \
793      table.o          \
794      timelocal.o     \
795      tolower.o        \
796      towlower.o      \
797      ungetwc.o       \
798      utf8.o           \
799      wcrctomb.o       \
800      wcscasecmp.o    \
801      wcscoll.o       \
802      wcsftime.o      \
803      wcsnrtombs.o    \
804      wcsrtombs.o     \
805      wcswidth.o      \
806      wcstombs.o      \
807      wcsxfrm.o       \
808      wctob.o         \
809      wctomb.o        \
810      wctrans.o       \
811      wctype.o        \
812      wcwidth.o       \
813      wscoll.o        \
\
815 AIOOBS= \
816      aio.o           \
817      aio_alloc.o    \
818      posix_aio.o    \
\
820 RTOBS= \
821      clock_timer.o  \
822      mqueue.o       \
823      pos4obj.o      \
824      sched.o        \
825      sem.o          \
826      shm.o          \
827      sigev_thread.o \
\
829 SECFLAGSOBS= \
830      secflags.o    \
\
832 TPOOLOBS= \
833      thread_pool.o \
\
835 THREADSOBS= \
836      alloc.o        \
837      assfail.o      \
838      cll_thr.o      \
839      cancel.o       \
840      door_calls.o   \
841      tmem.o         \
842      pthr_attr.o    \
843      pthr_barrier.o \
844      pthr_cond.o    \
845      pthr_mutex.o   \
846      pthr_rwlock.o  \
847      pthread.o      \
848      rwlock.o       \
849      scalls.o       \
850      sema.o         \
851      sigaction.o    \

```

```

852      spawn.o        \
853      synch.o         \
854      tdb_agent.o    \
855      thr.o           \
856      thread_interface.o \
857      tls.o          \
858      tsd.o          \
\
860 THREADSMACHOBS= \
861      machdep.o    \
\
863 THREADSASMOBS= \
864      asm_subr.o  \
\
866 UNICODEOBS= \
867      u8_textprep.o \
868      uconv.o     \
\
870 UNWINDMACHOBS= \
871      call_frame_inst.o \
872      eh_frame.o       \
873      thrp_unwind.o    \
874      unwind.o         \
\
876 pics/unwind.o:= COPTFLAG64 = \
\
878 UNWINDASMOBS= \
879      unwind_frame.o \
\
881 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
882 # This macro should ALWAYS be empty; native APIs are already 'large file'.
883 PORTSYS64= \
\
885 PORTSYS= \
886      _autofssys.o \
887      access.o     \
888      acctctl.o    \
889      bsd_signal.o \
890      chmod.o      \
891      chown.o      \
892      corectl.o    \
893      epoll.o      \
894      exaccts.o    \
895      execl.o      \
896      execl.o      \
897      execv.o      \
898      eventfd.o    \
899      fcntl.o      \
900      getpagesizes.o \
901      getpeerucred.o \
902      inst_sync.o   \
903      issetugid.o  \
904      label.o       \
905      link.o        \
906      lockf.o       \
907      lwp.o         \
908      lwp_cond.o    \
909      lwp_rwlock.o  \
910      lwp_sigmask.o \
911      meminfosys.o \
912      mkdir.o       \
913      mknod.o       \
914      msgsys.o      \
915      nfssys.o      \
916      open.o        \
917      pgrp.o        \

```

```

918     posix_sigwait.o  \
919     ppriv.o          \
920     psetsys.o       \
921     rctlsys.o       \
922     readlink.o      \
923     rename.o        \
924     sbrk.o           \
925     semsys.o         \
926     set_errno.o     \
927     sharefs.o       \
928     shmsys.o        \
929     sidsys.o        \
930     siginterrupt.o  \
931     signal.o         \
932     signalfd.o      \
933     sigpending.o    \
934     sigstack.o      \
935     stat.o           \
936     symlink.o       \
937     tasksys.o       \
938     time.o           \
939     time_util.o     \
940     timerfd.o       \
941     ucontext.o      \
942     unlink.o        \
943     ustat.o         \
944     utimesys.o     \
945     zone.o          \
\
947 PORTREGEX=        \
948     glob.o          \
949     regcmp.o        \
950     regcomp.o       \
951     regerror.o      \
952     regex.o         \
953     regexec.o       \
954     regfree.o       \
955     wordexp.o       \
\
957 VALUES=          \
958     values-Xa.o     \
\
960 MOSTOBSJS=        \
961     $(STRETS)       \
962     $(CRTOBSJS)     \
963     $(DYNOBJS)      \
964     $(FPOBJS)       \
965     $(I386FPOBJS)  \
966     $(FPASMOBJS)   \
967     $(ATOMICOBJS)  \
968     $(CHACHAOBJS)  \
969     $(XATTROBJS)   \
970     $(COMOBSJS)    \
971     $(GENOBSJS)    \
972     $(PORTFP)      \
973     $(PORTGEN)     \
974     $(PORTGEN64)   \
975     $(PORTI18N)    \
976     $(PORTI18N_COND) \
977     $(PORTINET)    \
978     $(PORTLOCALE)  \
979     $(PORTPRINT)   \
980     $(PORTPRINT_W) \
981     $(PORTREGEX)   \
982     $(PORTSTDIO)   \
983     $(PORTSTDIO64) \

```

```

984     $(PORTSTDIO_W)  \
985     $(PORTSYS)      \
986     $(PORTSYS64)    \
987     $(AIOOBSJS)    \
988     $(RTOBSJS)     \
989     $(SECFLAGSOBSJS) \
990     $(TPOOLOBSJS)  \
991     $(THREADSOBSJS) \
992     $(THREADSMACHOBSJS) \
993     $(THREADSASMOBSJS) \
994     $(UNICODEOBSJS) \
995     $(UNWINDMACHOBSJS) \
996     $(UNWINDASMOBSJS) \
997     $(COMSYSOBSJS)  \
998     $(SYSOBSJS)     \
999     $(COMSYSOBSJS64) \
1000    $(SYSOBSJS64)   \
1001    $(VALUES)       \
\
1003 TRACEOBSJS=      \
1004     plockstat.o    \
\
1006 # NOTE: libc.so.1 must be linked with the minimal crt1.o and crtn.o
1007 # modules whose source is provided in the $(SRC)/lib/crt directory.
1008 # This must be done because otherwise the Sun C compiler would insert
1009 # its own versions of these modules and those versions contain code
1010 # to call out to C++ initialization functions. Such C++ initialization
1011 # functions can call back into libc before thread initialization is
1012 # complete and this leads to segmentation violations and other problems.
1013 # Since libc contains no C++ code, linking with the minimal crt1.o and
1014 # crtn.o modules is safe and avoids the problems described above.
1015 OBJECTS= $(CRTI) $(MOSTOBSJS) $(CRTN)
1016 CRTSRCS= ../../crt/amd64

1018 # include common library definitions
1019 include ../../Makefile.lib
1020 include ../../Makefile.lib.64

1022 CFLAGS64 += $(CTF_FLAGS)

1024 # This is necessary to avoid problems with calling _ex_unwind().
1025 # We probably don't want any inlining anyway.
1026 CFLAGS64 += -xinline=

1028 CERRWARN += _gcc=-Wno-parentheses
1029 CERRWARN += _gcc=-Wno-switch
1030 CERRWARN += _gcc=-Wno-uninitialized
1031 CERRWARN += _gcc=-Wno-unused-value
1032 CERRWARN += _gcc=-Wno-unused-label
1033 CERRWARN += _gcc=-Wno-unused-variable
1034 CERRWARN += _gcc=-Wno-type-limits
1035 CERRWARN += _gcc=-Wno-char-subscripts
1036 CERRWARN += _gcc=-Wno-clobbered
1037 CERRWARN += _gcc=-Wno-unused-function
1038 CERRWARN += _gcc=-Wno-address

1040 $(RELEASE_BUILD)CERRWARN += _gcc=-Wno-unused

1042 # not linted
1043 SMATCH=off

1045 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1046 # enables ASSERT() checking in the threads portion of the library.
1047 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1048 THREAD_DEBUG =
1049 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

```

```

1051 # Make string literals read-only to save memory
1052 CFLAGS64 += $(XSTRCONST)

1054 ALTPICS= $(TRACEOBS:=pics/%)

1056 $(DYNLIB) := BUILD.SO = $(LD) -o $$@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1058 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1060 CPPFLAGS= -D_REENTRANT -D$(MACH64) -D_$(MACH64) $(THREAD_DEBUG) \
1061 -I. -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1062 ASFLAGS= $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) \
1063 $(amd64_AS_XARCH)

1065 # As a favor to the dtrace syscall provider, libc still calls the
1066 # old syscall traps that have been obsoleted by the *at() interfaces.
1067 # Delete this to compile libc using only the new *at() system call traps
1068 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1070 # proc64_id.o is built with defines in $(SRC)/uts/intel/sys/x86_archext.h
1071 pics/proc64_id.o := CFLAGS64 += -I$(SRC)/uts/intel

1073 # Inform the run-time linker about libc specialized initialization
1074 RTLDINFO = -z rtldinfo=tls_rtldinfo
1075 DYNFLAGS += $(RTLDINFO)

1077 # Force libc's internal references to be resolved immediately upon loading
1078 # in order to avoid critical region problems. Since almost all libc symbols
1079 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1080 DYNFLAGS += -znow

1082 BUILD.s= $(AS) $(ASFLAGS) $< -o $$@

1084 # Override this top level flag so the compiler builds in its native
1085 # C99 mode. This has been enabled to support the complex arithmetic
1086 # added to libc.
1087 CSTD= $(CSTD_GNU99)

1089 # libc method of building an archive
1090 # The "$(GREP) -v ' L '" part is necessary only until
1091 # lorder is fixed to ignore thread-local variables.
1092 BUILD.AR= $(RM) $$@ ; \
1093 $(AR) q $$@ $(LORDER) $(MOSTOBS:=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR)

1095 # extra files for the clean target
1096 CLEANFILES += \
1097 $(LIBCDIR)/port/gen/errlst.c \
1098 $(LIBCDIR)/port/gen/new_list.c \
1099 assym.h \
1100 genassym \
1101 crt/_rtld.s \
1102 pics/crti.o \
1103 pics/crtn.o \
1104 $(ALTPICS)

1106 CLOBBERFILES += $(LIB_PIC)

1108 # conditional assignments
1109 # $(DYNLIB) $(LIB_PIC) := DYNOBJS = _rtbootld.o
1110 $(DYNLIB) := CRTI = crt_i.o
1111 $(DYNLIB) := CRTN = crtn.o

1113 # Files which need the threads .il inline template
1114 TIL= \
1115 aio.o \

```

```

1116 alloc.o \
1117 assfail.o \
1118 atexit.o \
1119 atfork.o \
1120 cancel.o \
1121 door_calls.o \
1122 err.o \
1123 errno.o \
1124 lwp.o \
1125 ma.o \
1126 machdep.o \
1127 posix_aio.o \
1128 pthr_attr.o \
1129 pthr_barrier.o \
1130 pthr_cond.o \
1131 pthr_mutex.o \
1132 pthr_rwlock.o \
1133 pthread.o \
1134 rand.o \
1135 rwlock.o \
1136 scalls.o \
1137 sched.o \
1138 sema.o \
1139 sigaction.o \
1140 sigev_thread.o \
1141 spawn.o \
1142 stack.o \
1143 synch.o \
1144 tdb_agent.o \
1145 thr.o \
1146 thread_interface.o \
1147 thread_pool.o \
1148 thrp_unwind.o \
1149 tls.o \
1150 tmem.o \
1151 tsd.o \

1153 $(TIL:=pics/%) := CFLAGS64 += $(LIBCBASE)/threads/amd64.il

1155 # pics/mul64.o := CFLAGS64 += crt/mul64.il

1157 # large-file-aware components that should be built large

1159 #$(COMSYSOBS:=pics/%) := \
1160 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1162 #$(SYSOBS:=pics/%) := \
1163 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1165 #$(PORTGEN64:=pics/%) := \
1166 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1168 #$(PORTSTDIO64:=pics/%) := \
1169 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1171 #$(PORTSYS64:=pics/%) := \
1172 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1174 $(PORTSTDIO_W:=pics/%) := \
1175 CPPFLAGS += -D_WIDE

1177 $(PORTPRINT_W:=pics/%) := \
1178 CPPFLAGS += -D_WIDE

1180 $(PORTPRINT_C89:=pics/%) := \
1181 CPPFLAGS += -D_C89_INTMAX32

```

```
1183 $(PORTSTDIO_C89:%=pics/%) := \  
1184     CPPFLAGS += -D_C89_INTMAX32  
  
1186 $(PORTI18N_COND:%=pics/%) := \  
1187     CPPFLAGS += -D_WCS_LOGLONG  
  
1189 pics/arc4random.o :=     CPPFLAGS += -I$(SRC)/common/crypto/chacha  
  
1191 pics/__clock_gettime.o := CPPFLAGS += $(COMMPAGE_CPPFLAGS)  
  
1193 .KEEP_STATE:  
  
1195 all: $(LIBS) $(LIB_PIC)  
  
1197 # object files that depend on inline template  
1198 $(TIL:%=pics/%) : $(LIBCBASE)/threads/amd64.il  
1199 # pics/mul64.o: crt/mul64.il  
  
1201 # include common libc targets  
1202 include ../Makefile.targ  
  
1204 # We need to strip out all CTF data from the static library  
1205 $(LIB_PIC) := DIR = pics  
1206 $(LIB_PIC): pics $$ (PICS)  
1207     $(BUILD.AR)  
1208     $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&l  
1209     $(AR) -ts $@ > /dev/null  
1210     $(POST_PROCESS_A)  
  
1212 ASSYMDEP_OBJS= \\  
1213     _lwp_mutex_unlock.o \\  
1214     _stack_grow.o \\  
1215     asm_subr.o \\  
1216     getcontext.o \\  
1217     setjmp.o \\  
1218     tls_get_addr.o \\  
1219     vforkx.o  
  
1221 $(ASSYMDEP_OBJS:%=pics/%) : assym.h  
  
1223 # assym.h build rules  
  
1225 GENASSYM_C = genassym.c  
  
1227 genassym: $(GENASSYM_C)  
1228     $(NATIVECC) $(NATIVE_CFLAGS) -Iinc -I$(LIBCDIR)/inc $(CPPFLAGS.native) \  
1229     -o $@ $(GENASSYM_C)  
  
1231 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in  
  
1233 assym.h: $(OFFSETS) genassym  
1234     $(OFFSETS_CREATE) <$(OFFSETS) >$@  
1235     ./genassym >>$@  
  
1237 # derived C source and related explicit dependencies  
1238 $(LIBCDIR)/port/gen/errlst.c + \  
1239 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/  
1240     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist  
  
1242 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c  
  
1244 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```

*****
23639 Wed Apr 24 01:32:51 2019
new/usr/src/lib/libc/i386/Makefile.com
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2019 Joyent, Inc.
25 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
26 # Copyright 2013 Garrett D'Amore <garrett@damore.org>
27 # Copyright 2018 Nexenta Systems, Inc.
28 # Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
29 #
30 #
31 LIBCDIR=      $(SRC)/lib/libc
32 LIB_PIC=      libc_pic.a
33 VERS=        .1
34 CPP=         /usr/lib/cpp
35 TARGET_ARCH= i386
36 #
37 # include comm page definitions
38 include $(SRC)/lib/commpage/Makefile.shared.com
39 include $(SRC)/lib/commpage/Makefile.shared.targ
40 #
41 VALUES=      values-Xa.o
42 #
43 # objects are grouped by source directory
44 #
45 # local objects
46 STRETS=
47 #
48 CRTOBSJ=     \
49   cerror.o   \
50   cerror64.o
51 #
52 DYNOBJS=     \
53   _rtbootld.o
54 #
55 FPOBSJ=     \
56   _D_cplx_div.o   \
57   _D_cplx_div_ix.o \
58   _D_cplx_div_rx.o \
59   _D_cplx_lr_div.o \

```

```

60   _D_cplx_lr_div_ix.o \
61   _D_cplx_lr_div_rx.o \
62   _D_cplx_mul.o   \
63   _F_cplx_div.o   \
64   _F_cplx_div_ix.o \
65   _F_cplx_div_rx.o \
66   _F_cplx_lr_div.o \
67   _F_cplx_lr_div_ix.o \
68   _F_cplx_lr_div_rx.o \
69   _F_cplx_mul.o   \
70   _X_cplx_div.o   \
71   _X_cplx_div_ix.o \
72   _X_cplx_div_rx.o \
73   _X_cplx_lr_div.o \
74   _X_cplx_lr_div_ix.o \
75   _X_cplx_lr_div_rx.o \
76   _X_cplx_mul.o   \
77   fpgetmask.o    \
78   fpgetround.o   \
79   fpgetsticky.o  \
80   fpsetmask.o    \
81   fpsetround.o   \
82   fpsetsticky.o  \
83   fpstart.o      \
84   ieee.o
85 #
86 FPASMOBJS=    \
87   __xgetRD.o    \
88   _base_il.o    \
89   _xtoll.o      \
90   _xtoull.o     \
91   fpcw.o
92 #
93 ATOMICOBJS=   \
94   atomic.o
95 #
96 CHACHAOBJS=   \
97   chacha.o
98 #
99 XATTROBJS=    \
100  xattr_common.o
101 #
102 COMOBJS=      \
103  bcmp.o       \
104  bcopy.o      \
105  bsearch.o    \
106  bzero.o     \
107  qsort.o     \
108  strtol.o    \
109  strtoul.o   \
110  strtoll.o   \
111  strtoull.o
112 #
113 DTRACEOBSJ=   \
114  dtrace_data.o
115 #
116 SECFLAGSOBJS= \
117  secflags.o
118 #
119 GENOBJS=      \
120  $(COMMPAGE_OBJS) \
121  _div64.o       \
122  _divdi3.o     \
123  _getsp.o      \
124  _mul64.o      \
125  abs.o

```

```

126     alloca.o           \|
127     arc4random.o      \|
128     arc4random_uniform.o \|
129     byteorder.o      \|
130     byteorder64.o    \|
131     cuexit.o         \|
132     ecvt.o           \|
133     endian.o         \|
134     errlst.o         \|
135     i386_data.o      \|
136     ladd.o           \|
137     ldivide.o        \|
138     lmul.o           \|
139     lock.o           \|
140     lshift1.o        \|
141     lsign.o          \|
142     lsub.o           \|
143     makectxt.o       \|
144     memccpy.o        \|
145     memchr.o         \|
146     memcmp.o         \|
147     memcpy.o         \|
148     memset.o         \|
149     new_list.o       \|
150     setjmp.o         \|
151     siginfolst.o     \|
152     siglongjmp.o    \|
153     strcat.o         \|
154     strchr.o         \|
155     strcmp.o         \|
156     strcpy.o         \|
157     strlen.o         \|
158     strncat.o        \|
159     strncmp.o        \|
160     strncpy.o        \|
161     strnlen.o        \|
162     strrchr.o        \|
163     sync_instruction_memory.o \|

165 # sysobjs that contain large-file interfaces
166 COMSYSOBSJS64=
167     fstatvfs64.o     \|
168     getdents64.o    \|
169     getrlimit64.o   \|
170     lseek64.o       \|
171     mmap64.o        \|
172     pread64.o       \|
173     preadv64.o      \|
174     pwrite64.o      \|
175     pwritev64.o     \|
176     setrlimit64.o   \|
177     statvfs64.o     \|

179 SYSOBSJS64=

181 COMSYSOBSJS=
182     __clock_timer.o \|
183     __getloadavg.o  \|
184     __rusagesys.o   \|
185     __signotify.o   \|
186     __sigrt.o       \|
187     __time.o        \|
188     _lgrp_home_fast.o \|
189     _lgrpsys.o     \|
190     _nfssys.o       \|
191     _portfs.o       \|

```

```

192     _pset.o         \|
193     _rpcsys.o       \|
194     _sigaction.o    \|
195     _so_accept.o    \|
196     _so_bind.o      \|
197     _so_connect.o   \|
198     _so_getpeername.o \|
199     _so_getsockname.o \|
200     _so_getsockopt.o \|
201     _so_listen.o    \|
202     _so_recv.o      \|
203     _so_recvfrom.o  \|
204     _so_recvmsg.o   \|
205     _so_send.o      \|
206     _so_sendmsg.o   \|
207     _so_sendto.o    \|
208     _so_setsockopt.o \|
209     _so_shutdown.o  \|
210     _so_socket.o    \|
211     _so_socketpair.o \|
212     _sockconfig.o   \|
213     acct.o          \|
214     acl.o           \|
215     adjtime.o       \|
216     alarm.o         \|
217     brk.o           \|
218     chdir.o         \|
219     chroot.o        \|
220     cladm.o         \|
221     close.o         \|
222     execve.o        \|
223     exit.o          \|
224     facl.o          \|
225     fchdir.o        \|
226     fchroot.o       \|
227     fdsync.o        \|
228     fpathconf.o    \|
229     fstatfs.o       \|
230     fstatvfs.o      \|
231     getcpuuid.o     \|
232     getdents.o      \|
233     getegid.o       \|
234     geteuid.o       \|
235     getgid.o        \|
236     getgroups.o     \|
237     gethrtime.o     \|
238     getitimer.o     \|
239     getmsg.o         \|
240     getpid.o        \|
241     getpmsg.o       \|
242     getppid.o       \|
243     getrandom.o     \|
244     getrlimit.o     \|
245     getuid.o        \|
246     gtty.o          \|
247     install_utrap.o \|
248     ioctl.o         \|
249     kaio.o          \|
250     kill.o          \|
251     llseek.o        \|
252     lseek.o         \|
253     mmapobjsys.o    \|
254     memcntl.o       \|
255     mincore.o       \|
256     mmap.o          \|
257     modctl.o        \|

```

```

258     mount.o           \
259     mprotect.o        \
260     munmap.o          \
261     nice.o            \
262     ntp_adjtime.o     \
263     ntp_gettime.o     \
264     p_online.o        \
265     pathconf.o        \
266     pause.o           \
267     pcsample.o        \
268     pipe2.o           \
269     pollsys.o         \
270     pread.o           \
271     preadv.o          \
272     prioctlset.o      \
273     processor_bind.o  \
274     processor_info.o  \
275     profil.o          \
276     psecflagsset.o   \
277     putmsg.o          \
278     putpmsg.o         \
279     pwrite.o          \
280     pwritev.o         \
281     read.o            \
282     readv.o           \
283     resolvepath.o    \
284     seteguid.o        \
285     setgid.o          \
286     setgroups.o       \
287     setitimer.o       \
288     setreid.o         \
289     setrlimit.o       \
290     setuid.o          \
291     sigaltstk.o       \
292     sigprocmsk.o     \
293     sigsendset.o     \
294     sigsuspend.o     \
295     statfs.o          \
296     statvfs.o         \
297     stty.o            \
298     sync.o            \
299     sysconfig.o       \
300     sysfs.o           \
301     sysinfo.o         \
302     syslwp.o          \
303     times.o           \
304     ulimit.o          \
305     umask.o           \
306     umount2.o         \
307     utssys.o          \
308     uucopy.o          \
309     vhangup.o         \
310     waitid.o          \
311     write.o           \
312     writev.o         \
313     yield.o           \
\
315 SYSOBSJ=
316     __clock_gettime.o \
317     __clock_gettime_sys.o \
318     __getcontext.o    \
319     __uadmin.o        \
320     __lwp_mutex_unlock.o \
321     __stack_grow.o   \
322     door.o            \
323     forkx.o           \

```

```

324     forkallx.o        \
325     getcontext.o      \
326     gettimeofday.o   \
327     lwp_private.o     \
328     nuname.o          \
329     ptrace.o          \
330     syscall.o         \
331     sysi86.o          \
332     tls_get_addr.o    \
333     uadmin.o          \
334     umount.o          \
335     uname.o           \
336     vforkx.o          \
337     xstat.o           \
\
339 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
340 PORTGEN64=
341     _xftw64.o         \
342     attropen64.o     \
343     fts64.o           \
344     ftw64.o           \
345     mkstemp64.o      \
346     nftw64.o         \
347     tell64.o         \
348     truncate64.o     \
\
350 # objects from source under $(LIBCDIR)/port
351 PORTFP=
352     __flt_decim.o     \
353     __flt_rounds.o   \
354     __tbl_10_b.o     \
355     __tbl_10_h.o     \
356     __tbl_10_s.o     \
357     __tbl_2_b.o      \
358     __tbl_2_h.o      \
359     __tbl_2_s.o      \
360     __tbl_fdq.o      \
361     __tbl_tens.o     \
362     __x_power.o      \
363     _base_sup.o      \
364     aconvert.o        \
365     decimal_bin.o    \
366     double_decim.o   \
367     econvert.o        \
368     fconvert.o        \
369     file_decim.o     \
370     finite.o          \
371     fp_data.o         \
372     func_decim.o     \
373     gconvert.o        \
374     hex_bin.o         \
375     ieee_globals.o   \
376     pack_float.o     \
377     sigfpe.o          \
378     string_decim.o   \
\
380 PORTGEN=
381     _env_data.o       \
382     _xftw.o           \
383     a64l.o            \
384     abort.o           \
385     addsev.o          \
386     ascii_strcasecmp.o \
387     ascii_strncasecmp.o \
388     assert.o          \
389     atexit.o          \

```



```

390      atfork.o      \
391      atof.o       \
392      atoi.o       \
393      atol.o       \
394      atoll.o      \
395      attrat.o     \
396      attropen.o   \
397      basename.o   \
398      calloc.o     \
399      catgets.o    \
400      catopen.o    \
401      cfgetispeed.o \
402      cfgetospeed.o \
403      cfree.o      \
404      cfsetispeed.o \
405      cfsetospeed.o \
406      cftime.o     \
407      clock.o      \
408      closedir.o   \
409      closefrom.o  \
410      confstr.o    \
411      crypt.o      \
412      csetlen.o    \
413      ctime.o      \
414      ctime_r.o    \
415      daemon.o     \
416      default.o    \
417      directio.o  \
418      dirname.o    \
419      div.o        \
420      drand48.o    \
421      dup.o        \
422      env_data.o   \
423      err.o        \
424      errno.o      \
425      euclen.o     \
426      event_port.o \
427      execvp.o     \
428      explicit_bzero.o \
429      fattach.o    \
430      fdetach.o    \
431      fdopendir.o  \
432      ffs.o        \
433      flock.o      \
434      fls.o        \
435      fmtmsg.o     \
436      freezero.o   \
437      ftime.o      \
438      ftok.o       \
439      fts.o        \
440      ftw.o        \
441      gcvt.o       \
442      get_nprocs.o \
443      getauxv.o    \
444      getcwd.o     \
445      getdate_err.o \
446      getdthblsize.o \
447      getentropy.o \
448      getenv.o     \
449      getexecname.o \
450      getgrnam.o   \
451      getgrnam_r.o \
452      gethostid.o  \
453      gethostname.o \
454      gethz.o      \
455      getisax.o    \

```

```

456      getloadavg.o \
457      getlogin.o    \
458      getmntent.o   \
459      getnetgrent.o \
460      getopt.o      \
461      getopt_long.o \
462      getpagesize.o \
463      getpw.o       \
464      getpwnam.o    \
465      getpwnam_r.o  \
466      getrusage.o   \
467      getspent.o    \
468      getspent_r.o  \
469      getsubopt.o   \
470      gettxt.o      \
471      getusershell.o \
472      getut.o       \
473      getutx.o      \
474      getvfsent.o   \
475      getwd.o       \
476      getwidth.o    \
477      getxby_door.o \
478      gttx.o       \
479      hsearch.o     \
480      iconv.o       \
481      imaxabs.o     \
482      imaxdiv.o    \
483      index.o       \
484      initgroups.o  \
485      insque.o      \
486      isaexec.o     \
487      isastream.o   \
488      isatty.o      \
489      killpg.o      \
490      klpdlib.o     \
491      l64a.o        \
492      lckpwdf.o     \
493      lconstants.o  \
494      lexpl0.o      \
495      lfind.o       \
496      lfnt.o        \
497      lfnt_log.o    \
498      llabs.o       \
499      lldiv.o       \
500      llog10.o      \
501      lltostr.o     \
502      localtime.o  \
503      lsearch.o     \
504      madvise.o     \
505      malloc.o      \
506      memalign.o    \
507      memmem.o      \
508      memset_s.o    \
509      mkdev.o       \
510      mkdtemp.o     \
511      mkfifo.o      \
512      mkstemp.o     \
513      mktemp.o      \
514      mlock.o       \
515      mlockall.o   \
516      mon.o         \
517      msync.o       \
518      munlock.o     \
519      munlockall.o \
520      ndbm.o        \
521      nftw.o       \

```

```

522     nlspath_checks.o      \
523     nsparse.o             \
524     nss_common.o         \
525     nss_dbdefs.o         \
526     nss_deffinder.o      \
527     opendir.o            \
528     opt_data.o           \
529     perror.o             \
530     pfmt.o               \
531     pfmt_data.o          \
532     pfmt_print.o         \
533     pipe.o               \
534     plock.o              \
535     poll.o               \
536     posix_fadvise.o      \
537     posix_fallocate.o    \
538     posix_madvise.o      \
539     posix_memalign.o     \
540     priocntl.o           \
541     priv_str_xlate.o     \
542     privlib.o            \
543     psecflags.o          \
544     psiginfo.o           \
545     psignal.o            \
546     pt.o                 \
547     putpwent.o           \
548     putspent.o           \
549     raise.o              \
550     rand.o               \
551     random.o             \
552     rctlops.o            \
553     readdir.o            \
554     readdir_r.o          \
555     reallocarray.o       \
556     reallocarray.o       \
557     realpath.o           \
558     reboot.o             \
559     regexpr.o            \
560     remove.o             \
561     rewinddir.o          \
562     rindex.o             \
563     scandir.o            \
564     seekdir.o            \
565     select.o             \
566     set_constraint_handler_s.o \
567     setlabel.o           \
568     setpriority.o        \
569     settimeofday.o       \
570     sh_locks.o           \
571     sigflag.o            \
572     siglist.o            \
573     sigsend.o            \
574     sigsetops.o          \
575     signal.o             \
576     stack.o              \
577     stpcpy.o             \
578     stpncpy.o            \
579     str2sig.o            \
580     strcase_charmap.o    \
581     strchrnul.o          \
582     strcspn.o            \
583     strdup.o             \
584     strerror.o           \
585     strlcat.o            \
586     strlcpy.o            \
587     strndup.o            \

```

```

588     strpbrk.o            \
589     strsep.o             \
590     strsignal.o          \
591     strspn.o             \
592     strstr.o             \
593     strtod.o             \
594     strtointmax.o        \
595     strtok.o             \
596     strtok_r.o           \
597     strtonum.o           \
598     strtoumax.o          \
599     swab.o               \
600     swapctl.o            \
601     sysconf.o            \
602     syslog.o             \
603     tcdrain.o            \
604     tcflow.o             \
605     tcflush.o            \
606     tcgetattr.o          \
607     tcgetpgrp.o          \
608     tcgetsid.o           \
609     tcsendbreak.o        \
610     tcsetattr.o          \
611     tcsetpgrp.o          \
612     tell.o               \
613     telldir.o            \
614     tfind.o              \
615     time_data.o          \
616     time_gdata.o         \
617     timespec_get.o       \
618     tls_data.o           \
619     truncate.o           \
620     tsdalloc.o           \
621     tsearch.o            \
622     ttyname.o            \
623     ttyslot.o            \
624     ualarm.o             \
625     ucred.o              \
626     valloc.o             \
627     vlfmt.o              \
628     vpfmt.o              \
629     waitpid.o            \
630     walkstack.o          \
631     wdata.o              \
632     xgetwidth.o          \
633     xpg4.o               \
634     xpg6.o               \
636     PORTINET=           \
637         inet_lnaof.o      \
638         inet_makeaddr.o  \
639         inet_network.o    \
640         inet_ntoa.o       \
641         inet_ntop.o       \
642         inet_pton.o       \
644     PORTPRINT_W=        \
645         doprnt_w.o        \
647     PORTPRINT=         \
648         asprintf.o        \
649         doprnt.o          \
650         fprintf.o         \
651         printf.o          \
652         snprintf.o        \
653         sprintf.o         \

```

```

654      vfprintf.o          \
655      vprintf.o           \
656      vsnprintf.o        \
657      vsprintf.o         \
658      vwprintf.o         \
659      wprintf.o          \

661 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
662 PORTPRINT_C89=          \
663      vfprintf_c89.o     \
664      vprintf_c89.o      \
665      vsnprintf_c89.o    \
666      vsprintf_c89.o     \
667      vwprintf_c89.o     \

669 PORTSTDIO_C89=         \
670      vscanf_c89.o       \
671      vwscanf_c89.o      \

673 # portable stdio objects that contain large file interfaces.
674 # Note: fopen64 is a special case, as we build it small.
675 PORTSTDIO64=           \
676      fopen64.o          \
677      fpos64.o           \

679 PORTSTDIO_W=           \
680      doscan_w.o         \

682 PORTSTDIO=             \
683      __extensions.o     \
684      _endopen.o         \
685      _filbuf.o          \
686      _findbuf.o         \
687      _flsbuf.o         \
688      _wrtchk.o          \
689      clearerr.o         \
690      ctermid.o          \
691      ctermid_r.o        \
692      cuserid.o          \
693      data.o             \
694      doscan.o           \
695      fdopen.o           \
696      feof.o             \
697      ferrord.o          \
698      fgetc.o            \
699      fgets.o            \
700      fileno.o           \
701      flockf.o           \
702      flush.o            \
703      fopen.o            \
704      fpos.o             \
705      fputc.o            \
706      fputs.o            \
707      fread.o            \
708      fseek.o            \
709      fseeko.o           \
710      ftell.o            \
711      ftello.o           \
712      fwrite.o           \
713      getc.o             \
714      getchar.o          \
715      getline.o          \
716      getpass.o          \
717      gets.o             \
718      getw.o             \
719      mse.o              \

```

```

720      popen.o            \
721      putc.o             \
722      putchar.o         \
723      puts.o             \
724      putw.o            \
725      rewind.o          \
726      scanf.o           \
727      setbuf.o          \
728      setbuffer.o       \
729      setvbuf.o         \
730      system.o          \
731      tempnam.o         \
732      tmpfile.o         \
733      tmpnam_r.o        \
734      ungetc.o          \
735      vscanf.o          \
736      vwscanf.o         \
737      wscanf.o          \

739 PORTII18N=             \
740      getwchar.o         \
741      putwchar.o        \
742      putws.o           \
743      strtows.o         \
744      wcsnlen.o         \
745      wcsstr.o          \
746      wcstoimax.o       \
747      wcstol.o          \
748      wcstoul.o         \
749      wcswcs.o          \
750      wmemchr.o         \
751      wmemcmp.o         \
752      wmemcpy.o         \
753      wmemmove.o        \
754      wmemset.o         \
755      wscat.o           \
756      wchr.o            \
757      wscmp.o           \
758      wscopy.o          \
759      wscspn.o          \
760      wsdup.o           \
761      wslen.o           \
762      wsnecat.o         \
763      wsnncmp.o         \
764      wsnncpy.o         \
765      wspbrk.o          \
766      wsprintf.o        \
767      wsrchr.o          \
768      wsscanf.o         \
769      wsspn.o           \
770      wstod.o           \
771      wstok.o           \
772      wstol.o           \
773      wstoll.o          \
774      wsxfrm.o          \
775      gettext.o         \
776      gettext_gnu.o     \
777      gettext_real.o    \
778      gettext_util.o    \
779      plural_parser.o   \
780      wdresolve.o       \
781      _ctype.o          \
782      isascii.o         \
783      toascii.o         \

785 PORTII18N_COND=       \

```

```

786      wcstol_longlong.o  \
787      wcstoul_longlong.o

789 PORTLOCALE=           \
790      big5.o             \
791      btowc.o            \
792      collate.o          \
793      collcmp.o          \
794      euc.o              \
795      fnmatch.o          \
796      fgetwc.o           \
797      fgetws.o           \
798      fix_grouping.o     \
799      fputwc.o           \
800      fputws.o           \
801      fwide.o            \
802      gb18030.o          \
803      gb2312.o           \
804      gbk.o              \
805      getdate.o          \
806      isdigit.o          \
807      iswctype.o         \
808      ldpart.o           \
809      lmessages.o        \
810      lnumeric.o         \
811      lmonetary.o        \
812      localeconv.o       \
813      localeimpl.o       \
814      mbftowc.o          \
815      mblen.o            \
816      mbrlen.o           \
817      mbrtowc.o          \
818      mbsinit.o          \
819      mbsnrtowcs.o       \
820      mbsrtowcs.o        \
821      mbstowcs.o         \
822      mbtowc.o           \
823      mskanji.o          \
824      nextwctype.o       \
825      nl_langinfo.o      \
826      none.o             \
827      rune.o             \
828      runetype.o         \
829      setlocale.o        \
830      setrunelocale.o    \
831      strcasecmp.o       \
832      strcasestr.o       \
833      strcoll.o          \
834      strfmon.o          \
835      strftime.o         \
836      strncasecmp.o      \
837      strptime.o         \
838      strxfrm.o          \
839      table.o            \
840      timelocal.o        \
841      tolower.o          \
842      towlower.o         \
843      ungetwc.o          \
844      utf8.o             \
845      wcrtoomb.o         \
846      wcscasecmp.o       \
847      wcscoll.o          \
848      wcsftime.o         \
849      wcsnrtombs.o       \
850      wcsrtombs.o        \
851      wcswidth.o

```

```

852      wcstombs.o         \
853      wcsxfrm.o          \
854      wctob.o            \
855      wctomb.o           \
856      wctrans.o          \
857      wctype.o           \
858      wcwidth.o          \
859      wscoll.o

861 AIOOBSJ=                \
862      aio.o              \
863      aio_alloc.o        \
864      posix_aio.o

866 RTOBJS=                 \
867      clock_timer.o      \
868      mqueue.o           \
869      pos4obj.o          \
870      sched.o            \
871      sem.o              \
872      shm.o              \
873      sigev_thread.o

875 TPOOLOBSJ=              \
876      thread_pool.o

878 THREADSOBJS=            \
879      alloc.o            \
880      assfail.o          \
881      cancel.o           \
882      cll_thr.o          \
883      door_calls.o       \
884      tmem.o             \
885      pthr_attr.o        \
886      pthr_barrier.o     \
887      pthr_cond.o        \
888      pthr_mutex.o       \
889      pthr_rwlock.o      \
890      pthread.o          \
891      rwlock.o           \
892      scalls.o           \
893      sema.o             \
894      sigaction.o        \
895      spawn.o            \
896      synch.o            \
897      tdb_agent.o        \
898      thr.o              \
899      thread_interface.o \
900      tls.o              \
901      tsd.o

903 THREADSMACHOBSJ=       \
904      machdep.o

906 THREADSASMOBJS=        \
907      asm_subr.o

909 UNICODOBJS=             \
910      u8_textprep.o      \
911      uconv.o

913 UNWINDMACHOBSJ=        \
914      unwind.o

916 UNWINDASMOBJS=         \
917      unwind_frame.o

```

```

919 # objects that implement the transitional large file API
920 PORTSYS64= \
921     lockf64.o \
922     stat64.o

924 PORTSYS= \
925     _autofssys.o \
926     access.o \
927     acctctl.o \
928     bsd_signal.o \
929     chmod.o \
930     chown.o \
931     corectl.o \
932     epoll.o \
933     eventfd.o \
934     exacctsys.o \
935     execl.o \
936     execl.o \
937     execv.o \
938     fcntl.o \
939     getpagesizes.o \
940     getpeerucrd.o \
941     inst_sync.o \
942     issetugid.o \
943     label.o \
944     link.o \
945     lockf.o \
946     lwp.o \
947     lwp_cond.o \
948     lwp_rwlock.o \
949     lwp_sigmask.o \
950     meminfosys.o \
951     mkdir.o \
952     mknod.o \
953     msgsys.o \
954     nfssys.o \
955     open.o \
956     pgrpsys.o \
957     posix_sigwait.o \
958     ppriv.o \
959     psetsys.o \
960     rctlsys.o \
961     readlink.o \
962     rename.o \
963     sbrk.o \
964     semsys.o \
965     set_errno.o \
966     sharefs.o \
967     shmsys.o \
968     sidsys.o \
969     siginterrupt.o \
970     signal.o \
971     signalfd.o \
972     sigpending.o \
973     sigstack.o \
974     stat.o \
975     symlink.o \
976     tasksys.o \
977     time.o \
978     time_util.o \
979     timerfd.o \
980     ucontext.o \
981     unlink.o \
982     ustat.o \
983     utimesys.o \

```

```

984     zone.o

986 PORTREGEX= \
987     glob.o \
988     regcmp.o \
989     regcomp.o \
990     regerror.o \
991     regex.o \
992     regexec.o \
993     regfree.o \
994     wordexp.o

996 PORTREGEX64= \
997     glob64.o

999 MOSTOBSJ= \
1000     $(STRETS) \
1001     $(CRTOBSJ) \
1002     $(DYNOBJS) \
1003     $(FPOBJS) \
1004     $(FPASMOBJS) \
1005     $(ATOMICOBJS) \
1006     $(CHACHAOBJS) \
1007     $(XATTROBJS) \
1008     $(COMOBJS) \
1009     $(DTRACEOBJS) \
1010     $(GENOBJS) \
1011     $(PORTFP) \
1012     $(PORTGEN) \
1013     $(PORTGEN64) \
1014     $(PORTI18N) \
1015     $(PORTI18N_COND) \
1016     $(PORTINET) \
1017     $(PORTLOCALE) \
1018     $(PORTPRINT) \
1019     $(PORTPRINT_C89) \
1020     $(PORTPRINT_W) \
1021     $(PORTREGEX) \
1022     $(PORTREGEX64) \
1023     $(PORTSTDIO) \
1024     $(PORTSTDIO64) \
1025     $(PORTSTDIO_C89) \
1026     $(PORTSTDIO_W) \
1027     $(PORTSYS) \
1028     $(PORTSYS64) \
1029     $(AIOOBJS) \
1030     $(RTOBJS) \
1031     $(SECFLAGSOBJS) \
1032     $(TPOOLBJS) \
1033     $(THREADSOBJS) \
1034     $(THREADSMACHOBJS) \
1035     $(THREADSASMOBJS) \
1036     $(UNICODEOBJS) \
1037     $(UNWINDMACHOBJS) \
1038     $(UNWINDASMOBJS) \
1039     $(COMSYSOBJS) \
1040     $(YSOBJS) \
1041     $(COMSYSOBS64) \
1042     $(YSOBS64) \
1043     $(VALUES)

1045 TRACEOBSJ= \
1046     plockstat.o

1048 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
1049 # modules whose source is provided in the $(SRC)/lib/crt directory.

```

```

1050 # This must be done because otherwise the Sun C compiler would insert
1051 # its own versions of these modules and those versions contain code
1052 # to call out to C++ initialization functions. Such C++ initialization
1053 # functions can call back into libc before thread initialization is
1054 # complete and this leads to segmentation violations and other problems.
1055 # Since libc contains no C++ code, linking with the minimal crt1.o and
1056 # crtn.o modules is safe and avoids the problems described above.
1057 OBJECTS= $(CRTI) $(MOSTOBSJS) $(CRTN)
1058 CRTSRCS= ../../crt/i386

1060 LDPASS_OFF= $(POUND_SIGN)

1062 # include common library definitions
1063 include ../../Makefile.lib

1065 # we need to override the default SONAME here because we might
1066 # be building a variant object (still libc.so.1, but different filename)
1067 SONAME = libc.so.1

1069 CFLAGS += $(CCVERBOSE) $(CTF_FLAGS)

1071 # This is necessary to avoid problems with calling _ex_unwind().
1072 # We probably don't want any inlining anyway.
1073 XINLINE = -xinline=
1074 CFLAGS += $(XINLINE)

1076 CERRWARN += -_gcc=-Wno-parentheses
1077 CERRWARN += -_gcc=-Wno-switch
1078 CERRWARN += -_gcc=-Wno-uninitialized
1079 CERRWARN += -_gcc=-Wno-unused-value
1080 CERRWARN += -_gcc=-Wno-unused-label
1081 CERRWARN += -_gcc=-Wno-unused-variable
1082 CERRWARN += -_gcc=-Wno-type-limits
1083 CERRWARN += -_gcc=-Wno-char-subscripts
1084 CERRWARN += -_gcc=-Wno-clobbered
1085 CERRWARN += -_gcc=-Wno-unused-function
1086 CERRWARN += -_gcc=-Wno-address

1088 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused

1090 # not linted
1091 SMATCH=off

1093 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1094 # enables ASSERT() checking in the threads portion of the library.
1095 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1096 THREAD_DEBUG =
1097 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1099 # Make string literals read-only to save memory.
1100 CFLAGS += $(XSTRCONST)

1102 ALTPICS= $(TRACEOBSJS:=pics/%)

1104 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) \
1105 $(EXTPICS) $(LDLIBS)

1107 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1109 #
1110 # EXTN_CPPFLAGS and EXTN_CFLAGS set in enclosing Makefile
1111 #
1112 CFLAGS += $(EXTN_CFLAGS)
1113 CPPFLAGS= -D_REENTRANT -Di386 $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1114 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1115 ASFLAGS= $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) $(i386_AS_XARCH)

```

```

1117 # As a favor to the dtrace syscall provider, libc still calls the
1118 # old syscall traps that have been obsoleted by the *at() interfaces.
1119 # Delete this to compile libc using only the new *at() system call traps
1120 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1122 # Inform the run-time linker about libc specialized initialization
1123 RTLDINFO = -z rtldinfo=tls_rtldinfo
1124 DYNFLAGS += $(RTLDINFO)

1126 # Force libc's internal references to be resolved immediately upon loading
1127 # in order to avoid critical region problems. Since almost all libc symbols
1128 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1129 DYNFLAGS += -znw

1131 DYNFLAGS += -e __rtboot
1132 DYNFLAGS += $(EXTN_DYNFLAGS)

1134 # Inform the kernel about the initial DTrace area (in case
1135 # libc is being used as the interpreter / runtime linker).
1136 DTRACE_DATA = -zdtrace=dtrace_data
1137 DYNFLAGS += $(DTRACE_DATA)

1139 # DTrace needs an executable data segment.
1140 MAPFILE.NED=

1142 BUILD.s= $(AS) $(ASFLAGS) $< -o $@

1144 # Override this top level flag so the compiler builds in its native
1145 # C99 mode. This has been enabled to support the complex arithmetic
1146 # added to libc.
1147 CSTD= $(CSTD_GNU99)

1149 # libc method of building an archive
1150 # The "$(GREP) -v ' L '" part is necessary only until
1151 # lorder is fixed to ignore thread-local variables.
1152 BUILD.AR= $(RM) $@ ; \
1153 $(AR) q $@ '$(LORDER) $(MOSTOBSJS:=%$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR)

1155 # extra files for the clean target
1156 CLEANFILES += \
1157 $(LIBCDIR)/port/gen/errlst.c \
1158 $(LIBCDIR)/port/gen/new_list.c \
1159 assym.h \
1160 genassym \
1161 crt/_rtld.s \
1162 crt/_rtbootld.s \
1163 pics/_rtbootld.o \
1164 pics/crti.o \
1165 pics/crtn.o \
1166 $(ALTPICS)

1168 CLOBBERFILES += $(LIB_PIC)

1170 # conditional assignments
1171 $(DYNLIB) := CRTI = crt1.o
1172 $(DYNLIB) := CRTN = crtn.o

1174 # Files which need the threads .il inline template
1175 TIL= \
1176 aio.o \
1177 alloc.o \
1178 assfail.o \
1179 atexit.o \
1180 atfork.o \
1181 cancel.o

```

```

1182     door_calls.o      \
1183     err.o              \
1184     errno.o           \
1185     lwp.o              \
1186     ma.o               \
1187     machdep.o         \
1188     posix_aio.o       \
1189     pthr_attr.o       \
1190     pthr_barrier.o    \
1191     pthr_cond.o       \
1192     pthr_mutex.o     \
1193     pthr_rwlock.o    \
1194     pthread.o         \
1195     rand.o            \
1196     rwalk.o           \
1197     scalls.o          \
1198     sched.o           \
1199     sema.o            \
1200     sigaction.o       \
1201     sigev_thread.o   \
1202     spawn.o           \
1203     stack.o           \
1204     synch.o           \
1205     tdb_agent.o      \
1206     thr.o             \
1207     thread_interface.o \
1208     thread_pool.o    \
1209     tls.o             \
1210     tsd.o             \
1211     tmem.o            \
1212     unwind.o

```

```

1214 THREADS_INLINES = $(LIBCBASE)/threads/i386.il
1215 $(TIL:%=pics/%) := CFLAGS += $(THREADS_INLINES)

1217 # pics/mul64.o := CFLAGS += $(LIBCBASE)/crt/mul64.il

```

```

1219 # large-file-aware components that should be built large

```

```

1221 $(COMSYSOBS64:%=pics/%) := \
1222     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1224 $(SYSOBS64:%=pics/%) := \
1225     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1227 $(PORTGEN64:%=pics/%) := \
1228     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1230 $(PORTREGEX64:%=pics/%) := \
1231     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1233 $(PORTSTDIO64:%=pics/%) := \
1234     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1236 $(PORTSYS64:%=pics/%) := \
1237     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

```

```

1239 $(PORTSTDIO_W:%=pics/%) := \
1240     CPPFLAGS += -D_WIDE

```

```

1242 $(PORTPRINT_W:%=pics/%) := \
1243     CPPFLAGS += -D_WIDE

```

```

1245 $(PORTPRINT_C89:%=pics/%) := \
1246     CPPFLAGS += -D_C89_INTMAX32

```

```

1248 $(PORTSTDIO_C89:%=pics/%) := \
1249     CPPFLAGS += -D_C89_INTMAX32

```

```

1251 $(PORTI18N_COND:%=pics/%) := \
1252     CPPFLAGS += -D_WCS_LOGLONG

```

```

1254 pics/arc4random.o := CPPFLAGS += -I$(SRC)/common/crypto/chacha

```

```

1256 pics/__clock_gettime.o := CPPFLAGS += $(COMPAGE_CPPFLAGS)

```

```

1258 .KEEP_STATE:

```

```

1260 all: $(LIBS) $(LIB_PIC)

```

```

1262 # object files that depend on inline template

```

```

1263 $(TIL:%=pics/%) : $(LIBCBASE)/threads/i386.il

```

```

1264 # pics/mul64.o : $(LIBCBASE)/crt/mul64.il

```

```

1266 # include common libc targets

```

```

1267 include $(LIBCDIR)/Makefile.targ

```

```

1269 # We need to strip out all CTF and DOF data from the static library

```

```

1270 $(LIB_PIC) := DIR = pics

```

```

1271 $(LIB_PIC): pics $(PICS)

```

```

1272     $(BUILD.AR)

```

```

1273     $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1

```

```

1274     $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1

```

```

1275     $(AR) -ts $@ > /dev/null

```

```

1276     $(POST_PROCESS_A)

```

```

1278 $(LIBCBASE)/crt/_rtbootld.s : $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.o

```

```

1279     $(CC) $(CPPFLAGS) -_smatch=off $(CTF_FLAGS) -O -S $(C_PICFLAGS) \

```

```

1280         $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s

```

```

1281     $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@

```

```

1282     $(RM) $(LIBCBASE)/crt/_rtld.s

```

```

1284 # partially built from C source

```

```

1285 pics/_rtbootld.o : $(LIBCBASE)/crt/_rtbootld.s

```

```

1286     $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@

```

```

1287     $(CTFCONVERT_O)

```

```

1289 ASSYMDEP_OBJS=

```

```

1290     _lwp_mutex_unlock.o

```

```

1291     _stack_grow.o

```

```

1292     getcontext.o

```

```

1293     setjmp.o

```

```

1294     tls_get_addr.o

```

```

1295     vforkx.o

```

```

1297 $(ASSYMDEP_OBJS:%=pics/%) := CPPFLAGS += -I.

```

```

1299 $(ASSYMDEP_OBJS:%=pics/%) : assym.h

```

```

1301 # assym.h build rules

```

```

1303 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

```

```

1305 genassym: $(GENASSYM_C)

```

```

1306     $(NATIVECC) $(NATIVE_CFLAGS) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \

```

```

1307         -D__EXTENSIONS__ $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

```

```

1309 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

```

```

1311 assym.h: $(OFFSETS) genassym

```

```

1312     $(OFFSETS_CREATE) <$(OFFSETS) >$@

```

```

1313     ./genassym >>$@

```

```
1315 # derived C source and related explicit dependencies
1316 $(LIBCDIR)/port/gen/errlst.c + \
1317 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1318     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1320 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1322 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```



```

*****
4008 Wed Apr 24 01:32:52 2019
new/usr/src/lib/udapl/udapl_tavor/Makefile.com
10804 Enable unused warnings outside of uts
Reviewed by: Andy Stormont <astormont@racktopsystems.com>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 # Copyright 2019, Joyent, Inc.
26 # Copyright (c) 2018, Joyent, Inc.
27 LIBRARY=      udapl_tavor.a
28 VERS=         .1
29
30 LOCOBJS =     \
31     dapl_cno_create.o \
32     dapl_cno_free.o \
33     dapl_cno_modify_agent.o \
34     dapl_cno_query.o \
35     dapl_cno_util.o \
36     dapl_cno_wait.o \
37     dapl_cookie.o \
38     dapl_cr_accept.o \
39     dapl_cr_callback.o \
40     dapl_cr_handoff.o \
41     dapl_cr_query.o \
42     dapl_cr_reject.o \
43     dapl_cr_util.o \
44     dapl_debug.o \
45     dapl_ep_connect.o \
46     dapl_ep_create.o \
47     dapl_ep_create_with_srq.o \
48     dapl_ep_disconnect.o \
49     dapl_ep_dup_connect.o \
50     dapl_ep_free.o \
51     dapl_ep_get_status.o \
52     dapl_ep_modify.o \
53     dapl_ep_post_rdma_read.o \
54     dapl_ep_post_rdma_write.o \
55     dapl_ep_post_recv.o \
56     dapl_ep_post_send.o \
57     dapl_ep_query.o \
58     dapl_ep_reset.o \

```

```

59     dapl_ep_util.o \
60     dapl_evd_clear_unwaitable.o \
61     dapl_evd_connection_callb.o \
62     dapl_evd_cq_async_error_callb.o \
63     dapl_evd_create.o \
64     dapl_evd_dequeue.o \
65     dapl_evd_disable.o \
66     dapl_evd_dto_callb.o \
67     dapl_evd_enable.o \
68     dapl_evd_free.o \
69     dapl_evd_modify_cno.o \
70     dapl_evd_post_se.o \
71     dapl_evd_qp_async_error_callb.o \
72     dapl_evd_query.o \
73     dapl_evd_resize.o \
74     dapl_evd_set_unwaitable.o \
75     dapl_evd_un_async_error_callb.o \
76     dapl_evd_util.o \
77     dapl_evd_wait.o \
78     dapl_get_consumer_context.o \
79     dapl_get_handle_type.o \
80     dapl_hash.o \
81     dapl_hca_util.o \
82     dapl_ia_close.o \
83     dapl_ia_open.o \
84     dapl_ia_query.o \
85     dapl_ia_util.o \
86     dapl_init.o \
87     dapl_llist.o \
88     dapl_lmr_create.o \
89     dapl_lmr_free.o \
90     dapl_lmr_query.o \
91     dapl_lmr_sync_rdma.o \
92     dapl_lmr_util.o \
93     dapl_mr_util.o \
94     dapl_name_service.o \
95     dapl_osd.o \
96     dapl_provider.o \
97     dapl_psp_create.o \
98     dapl_psp_create_any.o \
99     dapl_psp_free.o \
100    dapl_psp_query.o \
101    dapl_pz_create.o \
102    dapl_pz_free.o \
103    dapl_pz_query.o \
104    dapl_pz_util.o \
105    dapl_ring_buffer_util.o \
106    dapl_rmr_bind.o \
107    dapl_rmr_create.o \
108    dapl_rmr_free.o \
109    dapl_rmr_query.o \
110    dapl_rmr_util.o \
111    dapl_rsp_create.o \
112    dapl_rsp_free.o \
113    dapl_rsp_query.o \
114    dapl_set_consumer_context.o \
115    dapl_sp_util.o \
116    dapl_srq.o \
117    dapl_srq_util.o
118
119 TAVOROBJS = \
120     dapl_tavor_hca.o \
121     dapl_tavor_hw.o \
122     dapl_arbel_hw.o \
123     dapl_hermon_hw.o \
124     dapl_tavor_ibtf_cm.o \

```

```
125     dapl_tavor_ibtf_dto.o \  
126     dapl_tavor_ibtf_mrsync.o \  
127     dapl_tavor_ibtf_gp.o \  
128     dapl_tavor_ibtf_util.o \  
129     dapl_tavor_wr.o  
  
131 OBJECTS = $(LOCOBJS) $(TAVOROBS)  
  
133 include $(SRC)/lib/Makefile.lib  
  
135 LIBS =      $(DYNLIB)  
136 LDLIBS +=   -ldevinfo -lsocket -lnsl -ldat -lc -ldladm  
  
138 SRCDIR =    ../common  
139 TAVORSRCDIR = ../tavor  
  
141 SRCS = $(LOCOBJS:%.o=$(SRCDIR)/%.c) $(TAVOROBS:%.o=$(TAVORSRCDIR)/%.c)  
  
143 CPPFLAGS += -I$(SRC)/lib/udapl/udapl_tavor/include  
144 CPPFLAGS += -I$(SRC)/lib/udapl/udapl_tavor/tavor  
145 CPPFLAGS += -I$(SRC)/uts/common/sys/ib/clients/daplt  
146 CPPFLAGS += -I$(SRC)/uts/common  
147 CPPFLAGS += -I$(SRC)/uts/common/sys/ib/clients  
148 CFLAGS +=   $(CCVERBOSE)  
149 LINTFLAGS += -DDAPL_DBG  
150 LINTFLAGS64 += -DDAPL_DBG  
  
152 CERRWARN += -_gcc=-Wno-parentheses  
153 CERRWARN += -_gcc=-Wno-uninitialized  
154 CERRWARN += -_gcc=-Wno-switch  
  
156 $(NOT_RELEASE_BUILD)CPPFLAGS += -DDAPL_DBG  
157 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused  
  
159 # not linted  
160 SMATCH=off  
  
159 $(NOT_RELEASE_BUILD)CPPFLAGS += -DDAPL_DBG  
  
162 .KEEP_STATE:  
  
164 all: $(LIBS)  
  
166 debug: all  
  
168 lint: lintcheck  
  
170 pics/%.o: $(TAVORSRCDIR)/%.c  
171     $(COMPILE.c) -o $@ $<  
172     $(POST_PROCESS_O)  
  
174 include $(SRC)/lib/Makefile.targ
```