```
**********************************************************
   35725 Thu Oct  4 22:48:33 2018
new/usr/src/Makefile.master
9868 unused cw translations should be removed
**********************************************************
```

```
   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 #

  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 #

  33 #
  34 # Makefile.master, global definitions for system source
  35 #
  36 ROOT=           /proto

  38 #
  39 # Adjunct root, containing an additional proto area to be used for headers
  40 # and libraries.
  41 #
  42 ADJUNCT_PROTO=

  44 #
  45 # Adjunct for building things that run on the build machine.
  46 #
  47 NATIVE_ADJUNCT= /usr

  49 #
  50 # RELEASE_BUILD should be cleared for final release builds.
  51 # NOT_RELEASE_BUILD is exactly what the name implies.
  52 #
  53 # __GNUC toggles the building of ON components using gcc and related tools.
  54 # Normally set to '#', set it to '' to do gcc build.
  55 #
  56 # The declaration POUND_SIGN is always '#'. This is needed to get around the
  57 # make feature that '#' is always a comment delimiter, even when escaped or
  58 # quoted. We use this macro expansion method to get POUND_SIGN rather than
  59 # always breaking out a shell because the general case can cause a noticable
  60 # slowdown in build times when so many Makefiles include Makefile.master.
  61 #
```



```
  62 # While the majority of users are expected to override the setting below
  63 # with an env file (via nightly or bldenv), if you aren't building that way
  64 # (ie, you're using "ws" or some other bootstrapping method) then you need
  65 # this definition in order to avoid the subshell invocation mentioned above.
  66 #

  68 PRE_POUND=                              pre\#
  69 POUND_SIGN=                             $(PRE_POUND:pre\%=%)

  71 NOT_RELEASE_BUILD=
  72 RELEASE_BUILD=                          $(POUND_SIGN)
  73 $(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
  74 PATCH_BUILD=                            $(POUND_SIGN)

  76 # SPARC_BLD is '#' for an Intel build.
  77 # INTEL_BLD is '#' for a Sparc build.
  78 SPARC_BLD_1=    $(MACH:i386=$(POUND_SIGN))
  79 SPARC_BLD=      $(SPARC_BLD_1:sparc=)
  80 INTEL_BLD_1=    $(MACH:sparc=$(POUND_SIGN))
  81 INTEL_BLD=      $(INTEL_BLD_1:i386=)

  83 # The variables below control the compilers used during the build.
  84 # There are a number of permutations.
  85 #
  86 # __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
  87 # one is not POUND_SIGN is the primary, with the other as the shadow.  They
  88 # may also be used to control entirely compiler-specific Makefile assignments.
  89 # __GNUC and GCC are the default.
  90 #
  91 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
  92 # There is no Sun C analogue.
  93 #
  94 # The following version-specific options are operative regardless of which
  95 # compiler is primary, and control the versions of the given compilers to be
  96 # used.  They also allow compiler-version specific Makefile fragments.
  97 #

  99 __SUNC=                 $(POUND_SIGN)
 100 $(__SUNC)__GNUC=        $(POUND_SIGN)
 101 __GNUC64=               $(__GNUC)

 103 # Allow build-time "configuration" to enable or disable some things.
 104 # The default is POUND_SIGN, meaning "not enabled". If the environment
 105 # passes in an override like ENABLE_SMB_PRINTING= (empty) that will
 106 # uncomment things in the lower Makefiles to enable the feature.
 107 ENABLE_SMB_PRINTING=    $(POUND_SIGN)

 109 # CLOSED is the root of the tree that contains source which isn't released
 110 # as open source
 111 CLOSED=         $(SRC)/../closed

 113 # BUILD_TOOLS is the root of all tools including compilers.
 114 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.

 116 BUILD_TOOLS=            /ws/onnv-tools
 117 ONBLD_TOOLS=            $(BUILD_TOOLS)/onbld

 119 # define runtime JAVA_HOME, primarily for cmd/pools/poold
 120 JAVA_HOME=      /usr/java
 121 # define buildtime JAVA_ROOT
 122 JAVA_ROOT=      /usr/java
 123 # Build uses java7 by default.  Pass one the variables below set to empty
 124 # string in the environment to override.
 125 BLD_JAVA_6=     $(POUND_SIGN)
 126 BLD_JAVA_8=     $(POUND_SIGN)
```

```
GNUC_ROOT=      /opt/gcc/4.4.4
GCCLIBDIR=      $(GNUC_ROOT)/lib
GCCLIBDIR64=    $(GNUC_ROOT)/lib/$(MACH64)

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

RPCGEN=         /usr/bin/rpcgen
STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
VTFONTCVT=      $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcvt
# echo(1) and true(1) are specified without absolute paths, so that the shell
# spawned by make(1) may use the built-in versions.  This is minimally
# problematic, as the shell spawned by make(1) is known and under control, the
# only risk being if the shell falls back to $PATH.
#
# We specifically want an echo(1) that does interpolation of escape sequences,
# which ksh93, /bin/sh, and bash will all provide.
ECHO=           echo
TRUE=           true
INS=            $(ONBLD_TOOLS)/bin/$(MACH)/install
SYMLINK=        /usr/bin/ln -s
LN=             /usr/bin/ln
MKDIR=          /usr/bin/mkdir
CHMOD=          /usr/bin/chmod
MV=             /usr/bin/mv -f
RM=             /usr/bin/rm -f
CUT=            /usr/bin/cut
NM=             /usr/ccs/bin/nm
DIFF=           /usr/bin/diff
GREP=           /usr/bin/grep
EGREP=          /usr/bin/egrep
ELFWRAP=        /usr/bin/elfwrap
KSH93=          /usr/bin/ksh93
SED=            /usr/bin/sed
AWK=            /usr/bin/nawk
CP=             /usr/bin/cp -f
MCS=            /usr/ccs/bin/mcs
CAT=            /usr/bin/cat
ELFDUMP=        /usr/ccs/bin/elfdump
M4=             /usr/bin/m4
STRIP=          /usr/ccs/bin/strip
LEX=            /usr/ccs/bin/lex
FLEX=           /usr/bin/flex
YACC=           /usr/ccs/bin/yacc
CPP=            /usr/lib/cpp
ANSI_CPP=       $(GNUC_ROOT)/bin/cpp
JAVAC=          $(JAVA_ROOT)/bin/javac
JAVAH=          $(JAVA_ROOT)/bin/javah
JAVADOC=        $(JAVA_ROOT)/bin/javadoc
RMIC=           $(JAVA_ROOT)/bin/rmic
JAR=            $(JAVA_ROOT)/bin/jar
CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
XREF=           $(ONBLD_TOOLS)/bin/xref
FIND=           /usr/bin/find
PERL=           /usr/bin/perl
PERL_VERSION=   5.10.0
PERL_PKGVERS=   -510
PERL_ARCH =             i86pc-solaris-64int
$(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
```



```
PYTHON_VERSION= 2.7
PYTHON_PKGVERS= -27
PYTHON=         /usr/bin/python$(PYTHON_VERSION)
SORT=           /usr/bin/sort
TOUCH=          /usr/bin/touch
WC=             /usr/bin/wc
XARGS=          /usr/bin/xargs
ELFEDIT=        /usr/bin/elfedit
DTRACE=         /usr/sbin/dtrace -xnolibs
UNIQ=           /usr/bin/uniq
TAR=            /usr/bin/tar
ASTBINDIR=      /usr/ast/bin
MSGCC=          $(ASTBINDIR)/msgcc
MSGFMT=         /usr/bin/msgfmt -s
LCDEF=          $(ONBLD_TOOLS)/bin/$(MACH)/localedef
TIC=            $(ONBLD_TOOLS)/bin/$(MACH)/tic
ZIC=            $(ONBLD_TOOLS)/bin/$(MACH)/zic
OPENSSL=        /usr/bin/openssl

FILEMODE=       644
DIRMODE=        755

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

# For stylistic checks
#
# Note that the X and C checks are not used at this time and may need
# modification when they are actually used.
#
CSTYLE=         $(ONBLD_TOOLS)/bin/cstyle
CSTYLE_TAIL=
HDRCHK=         $(ONBLD_TOOLS)/bin/hdrchk
HDRCHK_TAIL=
JSTYLE=         $(ONBLD_TOOLS)/bin/jstyle

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

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

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

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

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

# install a link
INSLINKTARGET=  $<
INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@
INS.symlink=    $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@
```

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

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

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

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

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

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

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

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

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

317 # Prevent the removal of static symbols by the SPARC code generator (cg).
318 # The x86 code generator (ube) does not remove such symbols and as such
319 # using this workaround is not applicable for x86.
320 #
321 CCSTATICSYM=            -Wc,-Qassembler-ounrefsym=0
322 #
323 # generate 32-bit addresses in the v9 kernel. Saves memory.
324 CCABS32=                -Wc,-xcode=abs32
325 #
```



```
326 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
327 # system calls.
328 CC32BITCALLERS=         -_gcc=-massume-32bit-callers

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

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

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

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

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

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

386 STAND_FLAGS_32 = $($(MACH)_STAND_FLAGS)
387 STAND_FLAGS_64 = $($(MACH64)_STAND_FLAGS)

389 #
390 # disable the incremental linker
391 ILDOFF=                 -xildoff
```

```
392 #
393 XDEPEND=                -xdepend
393 XFFLAG=                 -xF=%all
394 XESS=                   -xs
395 XSTRCONST=              -xstrconst

397 #
398 # turn warnings into errors (C)
399 CERRWARN = -errtags=yes -errwarn=%all
400 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
401 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

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

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

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

417 #
418 # turn warnings into errors (C++)
419 CCERRWARN=              -xwe

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

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

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

440 #
441 sparc_COPTFLAG=         -xO3
442 sparcv9_COPTFLAG=       -xO3
443 i386_COPTFLAG=          -O
444 amd64_COPTFLAG=         -xO3

446 COPTFLAG= $($(MACH)_COPTFLAG)
447 COPTFLAG64= $($(MACH64)_COPTFLAG)

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

453 # Direct the Sun Studio compiler to use a static globalization prefix based on t
454 # name of the module rather than something unique. Otherwise, objects
455 # will not build deterministically, as subsequent compilations of identical
456 # source will yeild objects that always look different.
```



```
457 #
458 # In the same spirit, this will also remove the date from the N_OPT stab.
459 CGLOBALSTATIC= -W0,-xglobalstatic

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

465 #
466 # Default debug format for Sun Studio 11 is dwarf, so force it to
467 # generate stabs.
468 #
469 DEBUGFORMAT=    -xdebugformat=stabs

471 #
472 # Flags used to build in debug mode for ctf generation.  Bugs in the Devpro
473 # compilers currently prevent us from building with cc-emitted DWARF.
474 #
475 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)
476 CTF_FLAGS_i386  = -g $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)

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

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

484 CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
485 CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
486 CTF_FLAGS       = $(CTF_FLAGS_32)

488 #
489 # Flags used with genoffsets
490 #
491 GOFLAGS = $(CALLSYMS) $(CDWARFSTR)

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

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

499 #
500 # tradeoff time for space (smaller is better)
501 #
502 sparc_SPACEFLAG         = -xspace -W0,-Lt
503 sparcv9_SPACEFLAG       = -xspace -W0,-Lt
504 i386_SPACEFLAG          = -xspace
505 amd64_SPACEFLAG         =

507 SPACEFLAG               = $($(MACH)_SPACEFLAG)
508 SPACEFLAG64             = $($(MACH64)_SPACEFLAG)

510 #
511 # The Sun Studio 11 compiler has changed the behaviour of integer
512 # wrap arounds and so a flag is needed to use the legacy behaviour
513 # (without this flag panics/hangs could be exposed within the source).
514 #
515 sparc_IROPTFLAG         = -W2,-xwrap_int
516 sparcv9_IROPTFLAG       = -W2,-xwrap_int
517 i386_IROPTFLAG          =
518 amd64_IROPTFLAG         =

520 IROPTFLAG               = $($(MACH)_IROPTFLAG)
521 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)
```

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

528 XREGSFLAG               = $($(MACH)_XREGSFLAG)
529 XREGSFLAG64             = $($(MACH64)_XREGSFLAG)

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

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

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

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

572 DTEXTDOM=-DTEXT_DOMAIN=\"$(TEXT_DOMAIN)\"       # For messaging.
573 DTS_ERRNO=-D_TS_ERRNO
574 CPPFLAGS.first= # Please keep empty.  Only lower makefiles should set this.
575 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
576         $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
577         $(ADJUNCT_PROTO:%=-I%/usr/include)
578 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
579                 $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
580 CPPFLAGS=       $(CPPFLAGS.first) $(CPPFLAGS.master)
581 AS_CPPFLAGS=    $(CPPFLAGS.first) $(CPPFLAGS.master)
582 JAVAFLAGS=      -source 1.6 -target 1.6 -Xlint:deprecation,-options

584 #
585 # For source message catalogue
586 #
587 .SUFFIXES: $(SUFFIXES) .i .po
588 MSGROOT= $(ROOT)/catalog
```



```
589 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
590 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
591 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
592 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

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

606 #
607 # This is overwritten by local Makefile when PROG is a list.
608 #
609 POFILE= $(PROG).po

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

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

633 CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
634 CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
635 CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
636                 $(CCUSERFLAGS)
637 CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
638                 $(CCUSERFLAGS64)

640 #
641 #
642 #
643 ELFWRAP_FLAGS   =
644 ELFWRAP_FLAGS64 =       -64

646 #
647 # Various mapfiles that are used throughout the build, and delivered to
648 # /usr/lib/ld.
649 #
650 MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
651 MAPFILE.NED_sparc =
652 MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
653 MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
654 MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
```

```
655 MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
656 MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy

658 #
659 # Generated mapfiles that are compiler specific, and used throughout the
660 # build.  These mapfiles are not delivered in /usr/lib/ld.
661 #
662 MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
663 $(__GNUC64)MAPFILE.NGB_sparc= \
664                         $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
665 MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
666 $(__GNUC64)MAPFILE.NGB_sparcv9= \
667                         $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
668 MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
669 $(__GNUC64)MAPFILE.NGB_i386= \
670                         $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
671 MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
672 $(__GNUC64)MAPFILE.NGB_amd64= \
673                         $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
674 MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

676 #
677 # A generic interface mapfile name, used by various dynamic objects to define
678 # the interfaces and interposers the object must export.
679 #
680 MAPFILE.INT =           mapfile-intf

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

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

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

715 CLASSPATH=      .
716 COMPILE.java=   $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

718 #
719 # Link time macros
720 #
```



```
721 CCNEEDED                = -lC
722 CCEXTNEEDED             = -lCrun -lCstd
723 $(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -lstdc++ -lgcc_s
724 $(__GNUC)CCEXTNEEDED    = $(CCNEEDED)

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

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

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

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

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

776 # Define native compilation macros
777 #

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

786 $(__GNUC)PRIMARY_CC=    gcc4,$(GNUC_ROOT)/bin/gcc,gnu
```

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

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


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

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

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

813 $(MACH)_CC=             $(BUILD_CC)
814 $(MACH)_CCC=            $(BUILD_CCC)
815 $(MACH)_CPP=            $(BUILD_CPP)
816 $(MACH)_LD=             $(BUILD_LD)
817 $(MACH)_LINT=           $(BUILD_LINT)
818 $(MACH64)_CC=           $(BUILD_CC)
819 $(MACH64)_CCC=          $(BUILD_CCC)
820 $(MACH64)_CPP=          $(BUILD_CPP)
821 $(MACH64)_LD=           $(BUILD_LD)
822 $(MACH64)_LINT=         $(BUILD_LINT)

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

827 i386_AS=               /usr/ccs/bin/as
828 $(__GNUC)i386_AS=      $(ONBLD_TOOLS)/bin/$(MACH)/aw
829 amd64_AS=              $(ONBLD_TOOLS)/bin/$(MACH)/aw

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

838 #
839 # Makefile.master.64 overrides these settings
840 #
841 CC=                     $(NATIVECC)
842 CCC=                    $(NATIVECCC)
843 CPP=                    $(NATIVECPP)
844 AS=                     $(NATIVEAS)
845 LD=                     $(NATIVELD)
846 LINT=                   $(NATIVELINT)

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

851 BDIRECT=        -Bdirect
852 BDYNAMIC=       -Bdynamic
```



```
853 BLOCAL=          -Blocal
854 BNODIRECT=       -Bnodirect
855 BREDUCE=         -Breduce
856 BSTATIC=         -Bstatic

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

880 GSHARED=         -G
881 CCMT=            -mt

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

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

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

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

908 AS_PICFLAGS=            $(C_PICFLAGS)
909 AS_BIGPICFLAGS=         $(C_BIGPICFLAGS)

911 #
912 # Default label for CTF sections
913 #
914 CTFCVTFLAGS=            -i -L VERSION

916 #
917 # Override to pass module-specific flags to ctfmerge.  Currently used only by
918 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
```

```
# stripping.
#
CTFMRGFLAGS=

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

# Rules (normally from make.rules) and macros which are used for post
# processing files. Normally, these do stripping of the comment section
# automatically.
#    RELEASE_CM:        Should be editted to reflect the release.
#    POST_PROCESS_O:    Post-processing for '.o' files.
#    POST_PROCESS_A:    Post-processing for '.a' files (currently null).
#    POST_PROCESS_SO:   Post-processing for '.so' files.
#    POST_PROCESS:      Post-processing for executable files (no suffix).
# Note that these macros are not completely generalized as they are to be
# used with the file name to be processed following.
#
# It is left as an exercise to Release Engineering to embellish the generation
# of the release comment string.
#
#       If this is a standard development build:
#               compress the comment section (mcs -c)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an installation build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an release build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#
# The following list of macros are used in the definition of RELEASE_CM
# which is used to label all binaries in the build:
#
#       RELEASE         Specific release of the build, eg: 5.2
#       RELEASE_MAJOR   Major version number part of $(RELEASE)
#       RELEASE_MINOR   Minor version number part of $(RELEASE)
#       VERSION         Version of the build (alpha, beta, Generic)
#       PATCHID         If this is a patch this value should contain
#                       the patchid value (eg: "Generic 100832-01"), otherwise
#                       it will be set to $(VERSION)
#       RELEASE_DATE    Date of the Release Build
#       PATCH_DATE      Date the patch was created, if this is blank it
#                       will default to the RELEASE_DATE
#
RELEASE_MAJOR=  5
RELEASE_MINOR=  11
RELEASE=        $(RELEASE_MAJOR).$(RELEASE_MINOR)
VERSION=        SunOS Development
PATCHID=        $(VERSION)
RELEASE_DATE=   release date not set
PATCH_DATE=     $(RELEASE_DATE)
RELEASE_CM=     "@($(POUND_SIGN))SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
DEV_CM=         "@($(POUND_SIGN))SunOS Internal Development: non-nightly build"

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

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

POST_PROCESS_O=
POST_PROCESS_A=
```



```
POST_PROCESS_SO=        $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)
POST_PROCESS=           $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)

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

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

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

#       Default build rules which perform comment section post-processing.
#
.c:
        $(LINK.c) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.c.o:
        $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
.c.a:
        $(COMPILE.c) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.s.o:
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_O)
.s.a:
        $(COMPILE.s) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.cc:
        $(LINK.cc) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.cc.o:
        $(COMPILE.cc) $(OUTPUT_OPTION) $<
        $(POST_PROCESS_O)
.cc.a:
        $(COMPILE.cc) -o $% $<
        $(AR) $(ARFLAGS) $@ $%
        $(PROCESS_COMMENT) $%
        $(RM) $%
.y:
        $(YACC.y) $<
        $(LINK.c) -o $@ y.tab.c $(LDLIBS)
        $(POST_PROCESS)
        $(RM) y.tab.c
```

```
.y.o:
        $(YACC.y) $<
        $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
        $(RM) y.tab.c
.l:
        $(RM) $*.c
        $(LEX.l) $< > $*.c
        $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
        $(POST_PROCESS)
        $(RM) $*.c
.l.o:
        $(RM) $*.c
        $(LEX.l) $< > $*.c
        $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
        $(RM) $*.c

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

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

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

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

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

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

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

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

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

.pl.po .pm.po:
        $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
        $(RM)   $@ ;
        $(SED) "/^domain/d" < $(<F).po > $@ ;
        $(RM) $(<F).po
```



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

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

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

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

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

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

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

.y.po:
        $(YACC) -d $<
        $(CPPFORPO) y.tab.c  > $<.i
        $(BUILD.po)
        $(RM) y.tab.c

.l.po:
        $(LEX) $<
        $(CPPFORPO) lex.yy.c  > $<.i
        $(BUILD.po)
        $(RM) lex.yy.c

#
# Rules to perform stylistic checks
#
.SUFFIXES: .x .xml .check .xmlchk

.h.check:
        $(DOT_H_CHECK)

.x.check:
        $(DOT_X_CHECK)

.xml.xmlchk:
        $(MANIFEST_CHECK)

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

```
**********************************************************
    2357 Thu Oct  4 22:48:34 2018
new/usr/src/cmd/ldapcachemgr/Makefile
9868 unused cw translations should be removed
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

PROG=   ldap_cachemgr

MANIFEST=       client.xml
SVCMETHOD=      ldap-client

include ../Makefile.cmd

#
# To compile in the $mgr SLP support
# 1. add -DSLP in to CPPFLAGS line
# 2. add cachemgr_discovery.o to the OBJS line
# 3. add -lslp in the LDLIBS line
#

ROOTLDAPLIB=    $(ROOT)/usr/lib/ldap
ROOTLDAPPROG=   $(PROG:%=$(ROOTLDAPLIB)/%)

ROOTMANIFESTDIR=        $(ROOTSVCNETWORKLDAP)

OBJS=   cachemgr.o cachemgr_getldap.o cachemgr_parse.o cachemgr_change.o

SRCS=   ${OBJS:%.o=%.c}

CPPFLAGS +=     -D_REENTRANT -DSUN_THREADS \
                -I$(SRC)/lib/libsldap/common \
                -I$(SRC)/lib/libldap5/include/ldap \
                -I$(SRC)/lib/libc/port/gen

CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-switch
CERRWARN +=     -_gcc=-Wno-uninitialized

# Message files
POFILE=         ldap_cachemgr.po
POFILES=        ${OBJS:%.o=%.po}
```



```
LINTOUT=        lint.out

# TCOV_FLAG=    -ql
# GPROF_FLAG=   -xpg
# DEBUG_FLAG=   -g

LDLIBS +=       -lumem -lsldap

# install macros and rule
#

.KEEP_STATE:

clean := TARGET= clean
clobber := TARGET= clobber

$(ROOTSVCMETHOD) := FILEMODE = 0555

all: $(PROG)

${PROG}: ${OBJS}
        ${LINK.c} ${OPT} -o $@ ${OBJS} ${LDLIBS}
        ${POST_PROCESS}

$(ROOTLDAPLIB):
        $(INS.dir)

$(ROOTLDAPLIB)/%:       %
        $(INS.file)

lint := LINTFLAGS=-x -b -u -h

lint:
        $(LINT.c) ${SRCS}  > $(LINTOUT) 2>&1

cstyle:
        ${CSTYLE} ${SRCS}

install: all $(ROOTLDAPLIB) $(ROOTLDAPPROG) $(ROOTMANIFEST) $(ROOTSVCMETHOD)

check:  $(CHKMANIFEST)

clean:
        $(RM) $(OBJS) $(LINTOUT)

$(POFILE): $(POFILES)
        $(RM) $@
        cat $(POFILES) > $@

include $(SRC)/cmd/Makefile.targ
```

```
**********************************************************
    2862 Thu Oct  4 22:48:34 2018
new/usr/src/cmd/nscd/Makefile
9868 unused cw translations should be removed
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1994, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Makefile for name service cache daemon
#

PROG=           nscd
MANIFEST=       name-service-cache.xml
SVCMETHOD=      svc-nscd

include ../Makefile.cmd

ROOTMANIFESTDIR=        $(ROOTSVCSYSTEM)

OBJS=   server.o getpw.o getgr.o gethost.o getnode.o \
        getether.o getrpc.o getproto.o getnet.o \
        getbootp.o getauth.o getserv.o \
        getnetmasks.o getprinter.o getproject.o \
        getexec.o getprof.o getuser.o cache.o \
        nscd_biggest.o nscd_wait.o \
        nscd_init.o nscd_access.o nscd_cfgfile.o nscd_config.o \
        nscd_dbimpl.o nscd_getentctx.o nscd_intaddr.o \
        nscd_log.o nscd_nswconfig.o nscd_nswstate.o nscd_nswcfgst.o \
        nscd_seqnum.o nscd_smfmonitor.o \
        nscd_switch.o nscd_nswparse.o nscd_initf.o nscd_selfcred.o \
        nscd_frontend.o nscd_admin.o nscd_door.o \
        gettnrhtp.o gettnrhdb.o

CLOBBERFILES=   nscd

SRCS=   ${OBJS:%.o=%.c}

CFLAGS +=       $(CCVERBOSE)
CPPFLAGS +=     -D_REENTRANT -DSUN_THREADS \
                -I../../lib/libc/port/gen -I../../lib/libc/inc \
                -I../../lib/libsldap/common
LINTFLAGS +=    -erroff=E_GLOBAL_COULD_BE_STATIC2
LINTFLAGS +=    -erroff=E_NAME_USED_NOT_DEF2
LINTFLAGS +=    -erroff=E_NAME_DEF_NOT_USED2

CERRWARN +=     -_gcc=-Wno-switch
```



```
CERRWARN +=     -_gcc=-Wno-uninitialized
CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-type-limits

# nscd interposes on many symbols, and must export others for its own dlsym()
# use, and dlsym() calls from libc.  Itemizing the interfaces within a mapfile
# is error-prone, so establish the whole object as an interposer.
LDFLAGS +=      $(ZINTERPOSE)

# TCOV_FLAG=    -ql
# GPROF_FLAG=   -xpg
# DEBUG_FLAG=   -g

PROGLIBS=       $(LDLIBS) -lresolv -lnsl -lsocket -lumem -lscf -lavl

# install macros and rule
#
ROOTPROG=       ${ROOTUSRSBIN}/nscd

.KEEP_STATE:

all: $(PROG) $(NISPROG)

${PROG}: ${OBJS}
        ${LINK.c} ${OPT} -o $@ ${OBJS} ${PROGLIBS}
        ${POST_PROCESS}

lint:
        $(LINT.c) ${SRCS} ${PROGLIBS}

cstyle:
        ${CSTYLE} ${SRCS}

install: all $(ROOTPROG) $(ROOTMANIFEST) $(ROOTSVCMETHOD)

check:  $(CHKMANIFEST)

clean:
        ${RM} ${OBJS}

${ROOTUSRSBIN}/%: %
        ${INS.file}

${ROOTUSRLIB}/%: %
        ${INS.file}

include ../Makefile.targ
```

```
**********************************************************
    6493 Thu Oct  4 22:48:35 2018
new/usr/src/lib/libnsl/Makefile.com
9868 unused cw translations should be removed
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1997, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2018 Nexenta Systems, Inc. All rights reserved.
#

LIBRARY= libnsl.a
VERS=   .1

# objects are listed by source directory

# common utility code used in more than one directory
COMMON=         common.o daemon_utils.o

DES=            des_crypt.o des_soft.o

DIAL=           dial.o

IPSEC=          algs.o

NETDIR=         netdir.o

NSS= \
gethostbyname_r.o gethostent.o gethostent_r.o gethostent6.o gethostby_door.o \
getipnodeby_door.o getipnodeby.o getrpcent.o  getrpcent_r.o inet_matchaddr.o \
netdir_inet.o netdir_inet_sundry.o \
parse.o getauthattr.o getprofattr.o getexecattr.o getuserattr.o getauuser.o

NETSELECT= netselect.o

NSL=  \
_conn_util.o    _data2.o        _errlst.o \
_utility.o      t_accept.o      t_alloc.o       t_bind.o        t_close.o \
t_connect.o     t_error.o       t_free.o        t_getinfo.o     t_getname.o \
t_getstate.o    t_listen.o      t_look.o        t_open.o        t_optmgmt.o \
t_rcv.o         t_rcvconnect.o  t_rcvdis.o      t_rcvrel.o      t_rcvudata.o \
t_rcvuderr.o    t_snd.o         t_snddis.o      t_sndrel.o      t_sndudata.o \
t_sndv.o        t_sndreldata.o  t_rcvv.o        t_rcvreldata.o  t_sysconf.o \
t_sndvudata.o   t_rcvvudata.o   t_sync.o        t_unbind.o      t_strerror.o \
xti_wrappers.o
```



```
WRAPPERS= \
tli_wrappers.o

RPC= \
auth_des.o      auth_none.o     auth_sys.o      auth_time.o     authdes_prot.o \
authsys_prot.o  can_use_af.o \
clnt_bcast.o    clnt_dg.o       clnt_door.o     clnt_generic.o  clnt_perror.o \
clnt_raw.o      clnt_simple.o   clnt_vc.o       fdsync.o        getdname.o \
key_call.o      key_prot.o      mt_misc.o \
netname.o       netnamer.o      pmap_clnt.o     pmap_prot.o \
rpc_callmsg.o   rpc_comdata.o   rpc_generic.o   rpc_prot.o rpc_sel2poll.o \
rpc_soc.o       rpc_td.o        rpcb_clnt.o     rpcb_prot.o \
rpcb_st_xdr.o   rpcdname.o      rpcsec_gss_if.o rtime_tli.o     svc.o \
svc_auth.o      svc_auth_loopb.o        svc_auth_sys.o  svc_dg.o \
svc_door.o      svc_generic.o   svc_raw.o       svc_run.o       svc_simple.o \
svc_vc.o        svcauth_des.o   svid_funcs.o    ti_opts.o       xdr.o \
xdr_array.o     xdr_float.o     xdr_mem.o       xdr_rec.o       xdr_refer.o \
xdr_sizeof.o    xdr_stdio.o

SAF= checkver.o  doconfig.o

YP=  \
dbm.o           yp_all.o        yp_b_clnt.o     yp_b_xdr.o      yp_bind.o  \
yp_enum.o       yp_master.o     yp_match.o      yp_order.o      yp_update.o \
yperr_string.o  yp_xdr.o        ypprot_err.o    ypupd.o \
yp_rsvd.o \
yppasswd_xdr.o

NIS_GEN=  \
nis_xdr.o       nis_subr.o      nis_misc.o          \
nis_misc_proc.o  nis_sec_mechs.o


NIS= $(NIS_GEN)

KEY= publickey.o xcrypt.o gen_dhkeys.o

OBJECTS= $(COMMON) $(DES) $(DIAL) $(IPSEC) $(NETDIR) $(NSS) $(NETSELECT) \
         $(NSL) $(WRAPPERS) $(RPC) $(SAF) $(YP) $(NIS) $(KEY)

# libnsl build rules
pics/%.o: ../common/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../des/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../dial/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../ipsec/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../netdir/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../nss/%.c
        $(COMPILE.c)  -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: ../netselect/%.c
```

```
128         $(COMPILE.c)  -o $@ $<
129         $(POST_PROCESS_O)

131 pics/%.o: ../nsl/%.c
132         $(COMPILE.c)  -o $@ $<
133         $(POST_PROCESS_O)

135 pics/%.o: ../rpc/%.c
136         $(COMPILE.c) -DPORTMAP -DNIS  -o $@ $<
137         $(POST_PROCESS_O)

139 pics/%.o: ../saf/%.c
140         $(COMPILE.c)  -o $@ $<
141         $(POST_PROCESS_O)

143 pics/%.o: ../yp/%.c
144         $(COMPILE.c)   -o $@ $<
145         $(POST_PROCESS_O)

147 pics/%.o: ../key/%.c
148         $(COMPILE.c)   -o $@ $<
149         $(POST_PROCESS_O)

151 pics/%.o: ../nis/gen/%.c ../nis/gen/nis_clnt.h
152         $(COMPILE.c) -o $@ $<
153         $(POST_PROCESS_O)


156 pics/%.o: ../nis/gen/nis_clnt.h
157         $(COMPILE.cc) -o $@ $<
158         $(POST_PROCESS_O)

160 # include library definitions
161 include ../../Makefile.lib

163 # install this library in the root filesystem
164 include ../../Makefile.rootfs

166 LIBS =          $(DYNLIB) $(LINTLIB)

168 SRCDIR=         ../common

170 # Override the position-independent code generation flags.
171 #
172 # These files are particularly rich with references to global things.
173 # Ordering is by number of got references per file of files that have
174 # non-performance sensitive code in them.
175 #
176 # If you need to add more files and the GOT overflows with "pic" items,
177 # then use the environment variable LD_OPTIONS=-Dgot,detail to have the
178 # linker print out the list of GOT hogs..

180 GOTHOGS =       dial.o print_obj.o clnt_perror.o nsl_stdio_prv.o netdir.o \
181                 algs.o netselect.o
182 BIGPICS =       $(GOTHOGS:%=pics/%)
183 $(BIGPICS) :=   sparc_C_PICFLAGS = $(C_BIGPICFLAGS)
184 $(BIGPICS) :=   i386_C_PICFLAGS = $(C_BIGPICFLAGS)

186 # Compile C++ code without exceptions to avoid a dependence on libC.
187 NOEXCEPTIONS= -noex
188 CCFLAGS += $(NOEXCEPTIONS)
189 CCFLAGS64 += $(NOEXCEPTIONS)

186 CPPFLAGS +=     -I$(SRC)/lib/libnsl/include -D_REENTRANT
187 CPPFLAGS +=     -I$(SRC)/lib/libnsl/dial
```



```
189 CFLAGS +=       $(CCVERBOSE)

191 # Make string literals read-only to save memory.
192 CFLAGS +=       $(XSTRCONST)
193 CFLAGS64 +=     $(XSTRCONST)
194 CCFLAGS +=      -_CC=-features=conststrings
195 CCFLAGS64 +=    -_CC=-features=conststrings

197 CERRWARN +=     -_gcc=-Wno-char-subscripts
198 CERRWARN +=     -_gcc=-Wno-parentheses
199 CERRWARN +=     -_gcc=-Wno-uninitialized
200 CERRWARN +=     -_gcc=-Wno-switch
201 CERRWARN +=     -_gcc=-Wno-char-subscripts
202 CERRWARN +=     -_gcc=-Wno-empty-body
203 CERRWARN +=     -_gcc=-Wno-unused-variable
204 CERRWARN +=     -_gcc=-Wno-clobbered

206 LIBMP =          -lmp
207 lint :=         LIBMP =
208 LDLIBS +=       $(LIBMP) -lmd -lc
209 DYNFLAGS +=     $(ZNODELETE)

211 $(LINTLIB):=    SRCS=$(SRCDIR)/$(LINTSRC)
212 LINTFLAGS +=    -m -DPORTMAP
213 LINTFLAGS64 +=  -m -DPORTMAP

215 .KEEP_STATE:

217 all: $(LIBS)

219 # Don't lint WRAPPERS as they are explicitly unclean
220 SRCS=   $(DES:%.o=../des/%.c)                    \
221         $(DIAL:%.o=../dial/%.c)                  \
222         $(IPSEC:%.o=../ipsec/%.c)                \
223         $(NETDIR:%.o=../netdir/%.c)              \
224         $(NSS:%.o=../nss/%.c)                    \
225         $(NETSELECT:%.o=../netselect/%.c)        \
226         $(NSL:%.o=../nsl/%.c)                    \
227         $(RPC:%.o=../rpc/%.c)                    \
228         $(SAF:%.o=../saf/%.c)                    \
229         $(YP:%.o=../yp/%.c)                      \
230         $(NIS_GEN:%.o=../nis/gen/%.c)            \
231         $(COMMON:%.o=../common/%.c)

233 lint:
234         @$(LINT.c) $(SRCS) $(LDLIBS)

236 # include library targets
237 include ../../Makefile.targ
```

```
**********************************************************
   41222 Thu Oct  4 22:48:35 2018
new/usr/src/tools/cw/cw.c
9868 unused cw translations should be removed
**********************************************************

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

/*
 * Copyright 2018, Richard Lowe.
 */
/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

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

/* If you modify this file, you must increment CW_VERSION */
#define CW_VERSION      "3.0"
#define CW_VERSION      "2.0"

/*
 * -#           Verbose mode
 * -###         Show compiler commands built by driver, no compilation
 * -A<name[(tokens)]>   Preprocessor predicate assertion
 * -B<[static|dynamic]> Specify dynamic or static binding
 * -C           Prevent preprocessor from removing comments
 * -c           Compile only - produce .o files, suppress linking
 * -cg92        Alias for -xtarget=ss1000
 * -D<name[=token]>     Associate name with token as if by #define
 * -d[y|n]      dynamic [-dy] or static [-dn] option to linker
 * -E           Compile source through preprocessor only, output to stdout
 * -erroff=<t>  Suppress warnings specified by tags t(%none, %all, <tag list>)
 * -errtags=<a> Display messages with tags a(no, yes)
 * -errwarn=<t> Treats warnings specified by tags t(%none, %all, <tag list>)
 *              as errors
 * -fast        Optimize using a selection of options
 * -fd          Report old-style function definitions and declarations
 * -features=zla        Allow zero-length arrays
 * -flags       Show this summary of compiler options
```



```
 * -fnonstd     Initialize floating-point hardware to non-standard preferences
 * -fns[=<yes|no>] Select non-standard floating point mode
 * -fprecision=<p> Set FP rounding precision mode p(single, double, extended)
 * -fround=<r>  Select the IEEE rounding mode in effect at startup
 * -fsimple[=<n>] Select floating-point optimization preferences <n>
 * -fsingle     Use single-precision arithmetic (-Xt and -Xs modes only)
 * -ftrap=<t>   Select floating-point trapping mode in effect at startup
 * -fstore      force floating pt. values to target precision on assignment
 * -G           Build a dynamic shared library
 * -g           Compile for debugging
 * -H           Print path name of each file included during compilation
 * -h <name>    Assign <name> to generated dynamic shared library
 * -I<dir>      Add <dir> to preprocessor #include file search path
 * -i           Passed to linker to ignore any LD_LIBRARY_PATH setting
 * -keeptmp     Keep temporary files created during compilation
 * -KPIC        Compile position independent code with 32-bit addresses
 * -Kpic        Compile position independent code
 * -L<dir>      Pass to linker to add <dir> to the library search path
 * -l<name>     Link with library lib<name>.a or lib<name>.so
 * -mc          Remove duplicate strings from .comment section of output files
 * -mr          Remove all strings from .comment section of output files
 * -mr,"string" Remove all strings and append "string" to .comment section
 * -mt          Specify options needed when compiling multi-threaded code
 * -native      Find available processor, generate code accordingly
 * -nofstore    Do not force floating pt. values to target precision
 *              on assignment
 * -nolib       Same as -xnolib
 * -noqueue     Disable queuing of compiler license requests
 * -norunpath   Do not build in a runtime path for shared libraries
 * -O           Use default optimization level (-xO2 or -xO3. Check man page.)
 * -o <outputfile> Set name of output file to <outputfile>
 * -P           Compile source through preprocessor only, output to .i  file
 * -PIC         Alias for -KPIC or -xcode=pic32
 * -p           Compile for profiling with prof
 * -pic         Alias for -Kpic or -xcode=pic13
 * -Q[y|n]      Emit/don't emit identification info to output file
 * -qp          Compile for profiling with prof
 * -R<dir[:dir]> Build runtime search path list into executable
 * -S           Compile and only generate assembly code (.s)
 * -s           Strip symbol table from the executable file
 * -t           Turn off duplicate symbol warnings when linking
 * -U<name>     Delete initial definition of preprocessor symbol <name>
 * -V           Report version number of each compilation phase
 * -v           Do stricter semantic checking
 * -W<c>,<arg>  Pass <arg> to specified component <c> (a,l,m,p,0,2,h,i,u)
 * -w           Suppress compiler warning messages
 * -Xa          Compile assuming ANSI C conformance, allow K & R extensions
 *              (default mode)
 * -Xc          Compile assuming strict ANSI C conformance
 * -Xs          Compile assuming (pre-ANSI) K & R C style code
 * -Xt          Compile assuming K & R conformance, allow ANSI C
 * -x386        Generate code for the 80386 processor
 * -x486        Generate code for the 80486 processor
 * -xarch=<a>   Specify target architecture instruction set
 * -xbuiltin[=<b>] When profitable inline, or substitute intrinisic functions
 *              for system functions, b={%all,%none}
 * -xCC         Accept C++ style comments
 * -xchar_byte_order=<o> Specify multi-char byte order <o> (default, high, low)
 * -xchip=<c>   Specify the target processor for use by the optimizer
 * -xcode=<c>   Generate different code for forming addresses
 * -xcrossfile[=<n>] Enable optimization and inlining across source files,
 *              n={0|1}
 * -xe          Perform only syntax/semantic checking, no code generation
 * -xF          Compile for later mapfile reordering or unused section
 *              elimination
 * -xhelp=<f>   Display on-line help information f(flags, readme, errors)
```

```
 * -xildoff     Cancel -xildon
 * -xildon      Enable use of the incremental linker, ild
 * -xinline=[<a>,...,<a>]  Attempt inlining of specified user routines,
 *              <a>={%auto,func,no%func}
 * -xlibmieee   Force IEEE 754 return values for math routines in
 *              exceptional cases
 * -xlibmil     Inline selected libm math routines for optimization
 * -xlic_lib=sunperf    Link in the Sun supplied performance libraries
 * -xlicinfo    Show license server information
 * -xM          Generate makefile dependencies
 * -xM1         Generate makefile dependencies, but exclude /usr/include
 * -xmaxopt=[off,1,2,3,4,5] maximum optimization level allowed on #pragma opt
 * -xnolib      Do not link with default system libraries
 * -xnolibmil   Cancel -xlibmil on command line
 * -xO<n>       Generate optimized code (n={1|2|3|4|5})
 * -xP          Print prototypes for function definitions
 * -xpentium    Generate code for the pentium processor
 * -xpg         Compile for profiling with gprof
 * -xprofile=<p> Collect data for a profile or use a profile to optimize
 *              <p>={{collect,use}[:<path>],tcov}
 * -xregs=<r>   Control register allocation
 * -xs          Allow debugging without object (.o) files
 * -xsb         Compile for use with the WorkShop source browser
 * -xsbfast     Generate only WorkShop source browser info, no compilation
 * -xsfpconst   Represent unsuffixed floating point constants as single
 *              precision
 * -xspace      Do not do optimizations that increase code size
 * -xstrconst   Place string literals into read-only data segment
 * -xtarget=<t> Specify target system for optimization
 * -xtemp=<dir> Set directory for temporary files to <dir>
 * -xtime       Report the execution time for each compilation phase
 * -xtransition Emit warnings for differences between K&R C and ANSI C
 * -xtrigraphs[=<yes|no>] Enable|disable trigraph translation
 * -xunroll=n   Enable unrolling loops n times where possible
 * -Y<c>,<dir>  Specify <dir> for location of component <c> (a,l,m,p,0,h,i,u)
 * -YA,<dir>    Change default directory searched for components
 * -YI,<dir>    Change default directory searched for include files
 * -YP,<dir>    Change default directory for finding libraries files
 * -YS,<dir>    Change default directory for startup object files
 */

/*
 * Translation table:
 */
/*
 * -#                           -v
 * -###                         error
 * -A<name[(tokens)]>           pass-thru
 * -B<[static|dynamic]>         pass-thru (syntax error for anything else)
 * -C                           pass-thru
 * -c                           pass-thru
 * -cg92                        -m32 -mcpu=v8 -mtune=supersparc (SPARC only)
 * -D<name[=token]>             pass-thru
 * -dy or -dn                   -Wl,-dy or -Wl,-dn
 * -E                           pass-thru
 * -erroff=E_EMPTY_TRANSLATION_UNIT ignore
 * -errtags=%all                -Wall
 * -errwarn=%all                -Werror else -Wno-error
 * -fast                        error
 * -fd                          error
 * -features=zla                ignore
 * -flags                       --help
 * -fnonstd                     error
 * -fns[=<yes|no>]              error
 * -fprecision=<p>              error
 * -fround=<r>                  error
```



```
 * -fsimple[=<n>]               error
 * -fsingle[=<n>]               error
 * -ftrap=<t>                   error
 * -fstore                      error
 * -G                           pass-thru
 * -g                           pass-thru
 * -H                           pass-thru
 * -h <name>                    pass-thru
 * -I<dir>                      pass-thru
 * -i                           pass-thru
 * -keeptmp                     -save-temps
 * -KPIC                        -fPIC
 * -Kpic                        -fpic
 * -L<dir>                      pass-thru
 * -l<name>                     pass-thru
 * -mc                          error
 * -mr                          error
 * -mr,"string"                 error
 * -mt                          -D_REENTRANT
 * -native                      error
 * -nofstore                    error
 * -nolib                       -nodefaultlibs
 * -noqueue                     ignore
 * -norunpath                   ignore
 * -O                           -O1 (Check the man page to be certain)
 * -o <outputfile>              pass-thru
 * -P                           -E -o filename.i (or error)
 * -PIC                         -fPIC (C++ only)
 * -p                           pass-thru
 * -pic                         -fpic (C++ only)
 * -Q[y|n]                      error
 * -qp                          -p
 * -R<dir[:dir]>                pass-thru
 * -S                           pass-thru
 * -s                           -Wl,-s
 * -t                           -Wl,-t
 * -U<name>                     pass-thru
 * -V                           --version
 * -v                           -Wall
 * -Wa,<arg>                    pass-thru
 * -Wp,<arg>                    pass-thru except -xc99=<a>
 * -Wl,<arg>                    pass-thru
 * -W{m,0,2,h,i,u>              error/ignore
 * -Wu,-xmodel=kernel           -ffreestanding -mcmodel=kernel -mno-red-zone
 * -xmodel=kernel               -ffreestanding -mcmodel=kernel -mno-red-zone
 * -Wu,-save_args               -msave-args
 * -w                           pass-thru
 * -Xa                          -std=iso9899:199409 or -ansi
 * -Xc                          -ansi -pedantic
 * -Xt                          error
 * -Xs                          -traditional -std=c89
 * -x386                        -march=i386 (x86 only)
 * -x486                        -march=i486 (x86 only)
 * -xarch=<a>                   table
 * -xbuiltin[=<b>]              -fbuiltin (-fno-builtin otherwise)
 * -xCC                         ignore
 * -xchar_byte_order=<o>        error
 * -xchip=<c>                   table
 * -xcode=<c>                   table
 * -xdebugformat=<format>       ignore (always use dwarf-2 for gcc)
 * -xcrossfile[=<n>]            ignore
 * -xe                          error
 * -xF                          error
 * -xhelp=<f>                   error
 * -xildoff                     ignore
 * -xildon                      ignore
```

```
 229  * -xinline                    ignore
 230  * -xlibmieee                  error
 231  * -xlibmil                    error
 232  * -xlic_lib=sunperf           error
 263  * -xM                         -M
 264  * -xM1                        -MM
 233  * -xmaxopt=[...]              error
 266  * -xnolib                     -nodefaultlibs
 267  * -xnolibmil                  error
 234  * -xO<n>                      -O<n>
 235  * -xP                         error
 270  * -xpentium                   -march=pentium (x86 only)
 271  * -xpg                        error
 236  * -xprofile=<p>               error
 237  * -xregs=<r>                  table
 238  * -xs                         error
 239  * -xsb                        error
 240  * -xsbfast                    error
 241  * -xsfpconst                  error
 242  * -xspace                     ignore (-not -Os)
 243  * -xstrconst                  ignore
 244  * -xtarget=<t>                table
 245  * -xtemp=<dir>                error
 246  * -xtime                      error
 247  * -xtransition                -Wtransition
 284  * -xtrigraphs=<yes|no>        -trigraphs -notrigraphs
 248  * -xunroll=n                  error
 249  * -W0,-xdbggen=no%usedonly    -fno-eliminate-unused-debug-symbols
 250  *                             -fno-eliminate-unused-debug-types
 251  * -Y<c>,<dir>                 error
 252  * -YA,<dir>                   error
 253  * -YI,<dir>                   -nostdinc -I<dir>
 254  * -YP,<dir>                   error
 255  * -YS,<dir>                   error
 256  */

 258 #include <ctype.h>
 259 #include <err.h>
 260 #include <errno.h>
 261 #include <fcntl.h>
 262 #include <getopt.h>
 263 #include <stdio.h>
 264 #include <stdlib.h>
 265 #include <string.h>
 266 #include <unistd.h>

 268 #include <sys/param.h>
 269 #include <sys/stat.h>
 270 #include <sys/types.h>
 271 #include <sys/utsname.h>
 272 #include <sys/wait.h>

 274 #define CW_F_CXX        0x01
 275 #define CW_F_SHADOW     0x02
 276 #define CW_F_EXEC       0x04
 277 #define CW_F_ECHO       0x08
 278 #define CW_F_XLATE      0x10
 279 #define CW_F_PROG       0x20

 281 typedef enum cw_op {
 282         CW_O_NONE = 0,
 283         CW_O_PREPROCESS,
 284         CW_O_COMPILE,
 285         CW_O_LINK
 286 } cw_op_t;
```

_____unchanged_portion_omitted_


```
 492 static void
 530 Xcmode(struct aelist *h)
 531 {
 532         static int xconce;

 534         if (xconce++)
 535                 return;

 537         newae(h, "-ansi");
 538         newae(h, "-pedantic-errors");
 539 }

 541 static void
 493 Xsmode(struct aelist *h)
 494 {
 495         static int xsonce;

 497         if (xsonce++)
 498                 return;

 500         newae(h, "-traditional");
 501         newae(h, "-traditional-cpp");
 502 }
```

_____unchanged_portion_omitted_

```
 566 static void
 567 do_gcc(cw_ictx_t *ctx)
 568 {
 569         int c;
 570         int pic = 0, nolibc = 0;
 571         int in_output = 0, seen_o = 0, c_files = 0;
 572         cw_op_t op = CW_O_LINK;
 573         char *model = NULL;
 574         char *nameflag;
 575         int     mflag = 0;

 577         if (ctx->i_flags & CW_F_PROG) {
 578                 newae(ctx->i_ae, "--version");
 579                 return;
 580         }

 582         newae(ctx->i_ae, "-fident");
 583         newae(ctx->i_ae, "-finline");
 584         newae(ctx->i_ae, "-fno-inline-functions");
 585         newae(ctx->i_ae, "-fno-builtin");
 586         newae(ctx->i_ae, "-fno-asm");
 587         newae(ctx->i_ae, "-fdiagnostics-show-option");
 588         newae(ctx->i_ae, "-nodefaultlibs");

 590 #if defined(__sparc)
 591         /*
 592          * The SPARC ldd and std instructions require 8-byte alignment of
 593          * their address operand.  gcc correctly uses them only when the
 594          * ABI requires 8-byte alignment; unfortunately we have a number of
 595          * pieces of buggy code that doesn't conform to the ABI.  This
 596          * flag makes gcc work more like Studio with -xmemalign=4.
 597          */
 598         newae(ctx->i_ae, "-mno-integer-ldd-std");
 599 #endif

 601         /*
 602          * This is needed because 'u' is defined
 603          * under a conditional on 'sun'.  Should
 604          * probably just remove the conditional,
 605          * or make it be dependent on '__sun'.
```

```
606          *
607          * -Dunix is also missing in enhanced ANSI mode
608          */
609         newae(ctx->i_ae, "-D__sun");

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

614         /*
615          * Walk the argument list, translating as we go ..
616          */
617         while (--ctx->i_oldargc > 0) {
618                 char *arg = *++ctx->i_oldargv;
619                 size_t arglen = strlen(arg);

621                 if (*arg == '-') {
622                         arglen--;
623                 } else {
624                         /*
625                          * Discard inline files that gcc doesn't grok
626                          */
627                         if (!in_output && arglen > 3 &&
628                             strcmp(arg + arglen - 3, ".il") == 0)
629                                 continue;

631                         if (!in_output && arglen > 2 &&
632                             arg[arglen - 2] == '.' &&
633                             (arg[arglen - 1] == 'S' || arg[arglen - 1] == 's' ||
634                             arg[arglen - 1] == 'c' || arg[arglen - 1] == 'i'))
635                                 c_files++;

637                         /*
638                          * Otherwise, filenames and partial arguments
639                          * are passed through for gcc to chew on.  However,
640                          * output is always discarded for the secondary
641                          * compiler.
642                          */
643                         if ((ctx->i_flags & CW_F_SHADOW) && in_output)
644                                 newae(ctx->i_ae, ctx->i_discard);
645                         else
646                                 newae(ctx->i_ae, arg);
647                         in_output = 0;
648                         continue;
649                 }

651                 if (ctx->i_flags & CW_F_CXX) {
652                         if (strncmp(arg, "-_g++=", 6) == 0) {
653                                 newae(ctx->i_ae, strchr(arg, '=') + 1);
654                                 continue;
655                         }
656                         if (strncmp(arg, "-compat=", 8) == 0) {
657                                 /* discard -compat=4 and -compat=5 */
658                                 continue;
659                         }
660                         if (strcmp(arg, "-Qoption") == 0) {
661                                 /* discard -Qoption and its two arguments */
662                                 if (ctx->i_oldargc < 3)
663                                         error(arg);
664                                 ctx->i_oldargc -= 2;
665                                 ctx->i_oldargv += 2;
666                                 continue;
667                         }
668                         if (strcmp(arg, "-xwe") == 0) {
669                                 /* turn warnings into errors */
670                                 newae(ctx->i_ae, "-Werror");
671                                 continue;
```



```
672                         }
722                         if (strcmp(arg, "-noex") == 0) {
723                                 /* no exceptions */
724                                 newae(ctx->i_ae, "-fno-exceptions");
725                                 /* no run time type descriptor information */
726                                 newae(ctx->i_ae, "-fno-rtti");
727                                 continue;
728                         }
729                         if (strcmp(arg, "-pic") == 0) {
730                                 newae(ctx->i_ae, "-fpic");
731                                 pic = 1;
732                                 continue;
733                         }
734                         if (strcmp(arg, "-PIC") == 0) {
735                                 newae(ctx->i_ae, "-fPIC");
736                                 pic = 1;
737                                 continue;
738                         }
673                         if (strcmp(arg, "-norunpath") == 0) {
674                                 /* gcc has no corresponding option */
675                                 continue;
676                         }
677                         if (strcmp(arg, "-nolib") == 0) {
678                                 /* -nodefaultlibs is on by default */
679                                 nolibc = 1;
680                                 continue;
681                         }
682 #if defined(__sparc)
683                         if (strcmp(arg, "-cg92") == 0) {
684                                 mflag |= xlate_xtb(ctx->i_ae, "v8");
685                                 xlate(ctx->i_ae, "super", xchip_tbl);
686                                 continue;
687                         }
688 #endif  /* __sparc */
689                 }

691                 switch ((c = arg[1])) {
692                 case '_':
693                         if ((strncmp(arg, nameflag, strlen(nameflag)) == 0) ||
694                             (strncmp(arg, "-_gcc=", 6) == 0) ||
695                             (strncmp(arg, "-_gnu=", 6) == 0)) {
696                                 newae(ctx->i_ae, strchr(arg, '=') + 1);
697                         }
698                         break;
699                 case '#':
700                         if (arglen == 1) {
701                                 newae(ctx->i_ae, "-v");
702                                 break;
703                         }
704                         error(arg);
705                         break;
706                 case 'g':
707                         newae(ctx->i_ae, "-gdwarf-2");
708                         break;
709                 case 'E':
710                         if (arglen == 1) {
711                                 newae(ctx->i_ae, "-xc");
712                                 newae(ctx->i_ae, arg);
713                                 op = CW_O_PREPROCESS;
714                                 nolibc = 1;
715                                 break;
716                         }
717                         error(arg);
718                         break;
719                 case 'c':
720                 case 'S':
```

```
721                         if (arglen == 1) {
722                                 op = CW_O_COMPILE;
723                                 nolibc = 1;
724                         }
725                         /* FALLTHROUGH */
726                 case 'C':
727                 case 'H':
728                 case 'p':
729                         if (arglen == 1) {
730                                 newae(ctx->i_ae, arg);
731                                 break;
732                         }
733                         error(arg);
734                         break;
735                 case 'A':
736                 case 'h':
737                 case 'I':
738                 case 'i':
739                 case 'L':
740                 case 'l':
741                 case 'R':
742                 case 'U':
743                 case 'u':
744                 case 'w':
745                         newae(ctx->i_ae, arg);
746                         break;
747                 case 'o':
748                         seen_o = 1;
749                         if (arglen == 1) {
750                                 in_output = 1;
751                                 newae(ctx->i_ae, arg);
752                         } else if (ctx->i_flags & CW_F_SHADOW) {
753                                 newae(ctx->i_ae, "-o");
754                                 newae(ctx->i_ae, ctx->i_discard);
755                         } else {
756                                 newae(ctx->i_ae, arg);
757                         }
758                         break;
759                 case 'D':
760                         newae(ctx->i_ae, arg);
761                         /*
762                          * XXX  Clearly a hack ... do we need _KADB too?
763                          */
764                         if (strcmp(arg, "-D_KERNEL") == 0 ||
765                             strcmp(arg, "-D_BOOT") == 0)
766                                 newae(ctx->i_ae, "-ffreestanding");
767                         break;
768                 case 'd':
769                         if (arglen == 2) {
770                                 if (strcmp(arg, "-dy") == 0) {
771                                         newae(ctx->i_ae, "-Wl,-dy");
772                                         break;
773                                 }
774                                 if (strcmp(arg, "-dn") == 0) {
775                                         newae(ctx->i_ae, "-Wl,-dn");
776                                         break;
777                                 }
778                         }
779                         if (strcmp(arg, "-dalign") == 0) {
780                                 /*
781                                  * -dalign forces alignment in some cases;
782                                  * gcc does not need any flag to do this.
783                                  */
784                                 break;
785                         }
786                         error(arg);
```



```
787                         break;
788                 case 'e':
789                         if (strcmp(arg,
790                             "-erroff=E_EMPTY_TRANSLATION_UNIT") == 0) {
791                                 /*
792                                  * Accept but ignore this -- gcc doesn't
793                                  * seem to complain about empty translation
794                                  * units
795                                  */
796                                 break;
797                         }
798                         /* XX64 -- ignore all -erroff= options, for now */
799                         if (strncmp(arg, "-erroff=", 8) == 0)
800                                 break;
801                         if (strcmp(arg, "-errtags=yes") == 0) {
802                                 warnings(ctx->i_ae);
803                                 break;
804                         }
805                         if (strcmp(arg, "-errwarn=%all") == 0) {
806                                 newae(ctx->i_ae, "-Werror");
807                                 break;
808                         }
809                         error(arg);
810                         break;
877                 case 'f':
878                         if (strcmp(arg, "-flags") == 0) {
879                                 newae(ctx->i_ae, "--help");
880                                 break;
881                         }
882                         if (strncmp(arg, "-features=zla", 13) == 0) {
883                                 /*
884                                  * Accept but ignore this -- gcc allows
885                                  * zero length arrays.
886                                  */
887                                 break;
888                         }
889                         error(arg);
890                         break;
811                 case 'G':
812                         newae(ctx->i_ae, "-shared");
813                         nolibc = 1;
814                         break;
815                 case 'k':
816                         if (strcmp(arg, "-keeptmp") == 0) {
817                                 newae(ctx->i_ae, "-save-temps");
818                                 break;
819                         }
820                         error(arg);
821                         break;
822                 case 'K':
823                         if (arglen == 1) {
824                                 if ((arg = *++ctx->i_oldargv) == NULL ||
825                                     *arg == '\0')
826                                         error("-K");
827                                 ctx->i_oldargc--;
828                         } else {
829                                 arg += 2;
830                         }
831                         if (strcmp(arg, "pic") == 0) {
832                                 newae(ctx->i_ae, "-fpic");
833                                 pic = 1;
834                                 break;
835                         }
836                         if (strcmp(arg, "PIC") == 0) {
837                                 newae(ctx->i_ae, "-fPIC");
838                                 pic = 1;
```

```
839                             break;
840                     }
841                     error("-K");
842                     break;
843             case 'm':
844                     if (strcmp(arg, "-mt") == 0) {
845                             newae(ctx->i_ae, "-D_REENTRANT");
846                             break;
847                     }
848                     if (strcmp(arg, "-m64") == 0) {
849                             newae(ctx->i_ae, "-m64");
850 #if defined(__x86)
851                             newae(ctx->i_ae, "-mtune=opteron");
852 #endif
853                             mflag |= M64;
854                             break;
855                     }
856                     if (strcmp(arg, "-m32") == 0) {
857                             newae(ctx->i_ae, "-m32");
858                             mflag |= M32;
859                             break;
860                     }
861                     error(arg);
862                     break;
863             case 'B':       /* linker options */
864             case 'M':
865             case 'z':
866                     {
867                             char *opt;
868                             size_t len;
869                             char *s;

871                             if (arglen == 1) {
872                                     opt = *++ctx->i_oldargv;
873                                     if (opt == NULL || *opt == '\0')
874                                             error(arg);
875                                     ctx->i_oldargc--;
876                             } else {
877                                     opt = arg + 2;
878                             }
879                             len = strlen(opt) + 7;
880                             if ((s = malloc(len)) == NULL)
881                                     nomem();
882                             (void) snprintf(s, len, "-Wl,-%c%s", c, opt);
883                             newae(ctx->i_ae, s);
884                             free(s);
885                     }
886                     break;
967             case 'n':
968                     if (strcmp(arg, "-noqueue") == 0) {
969                             /*
970                              * Horrid license server stuff - n/a
971                              */
972                             break;
973                     }
974                     error(arg);
975                     break;
887             case 'O':
888                     if (arglen == 1) {
889                             newae(ctx->i_ae, "-O");
890                             break;
891                     }
892                     error(arg);
893                     break;
894             case 'P':
895                     /*
```



```
896                      * We could do '-E -o filename.i', but that's hard,
897                      * and we don't need it for the case that's triggering
898                      * this addition.  We'll require the user to specify
899                      * -o in the Makefile.  If they don't they'll find out
900                      * in a hurry.
901                      */
902                     newae(ctx->i_ae, "-E");
903                     op = CW_O_PREPROCESS;
904                     nolibc = 1;
905                     break;
995             case 'q':
996                     if (strcmp(arg, "-qp") == 0) {
997                             newae(ctx->i_ae, "-p");
998                             break;
999                     }
1000                    error(arg);
1001                    break;
906             case 's':
907                     if (arglen == 1) {
908                             newae(ctx->i_ae, "-Wl,-s");
909                             break;
910                     }
911                     error(arg);
912                     break;
913             case 't':
914                     if (arglen == 1) {
915                             newae(ctx->i_ae, "-Wl,-t");
916                             break;
917                     }
918                     error(arg);
919                     break;
920             case 'V':
921                     if (arglen == 1) {
922                             ctx->i_flags &= ~CW_F_ECHO;
923                             newae(ctx->i_ae, "--version");
924                             break;
925                     }
926                     error(arg);
927                     break;
928             case 'v':
929                     if (arglen == 1) {
930                             warnings(ctx->i_ae);
931                             break;
932                     }
933                     error(arg);
934                     break;
935             case 'W':
936                     if (strncmp(arg, "-Wp,-xc99", 9) == 0) {
937                             /*
938                              * gcc's preprocessor will accept c99
939                              * regardless, so accept and ignore.
940                              */
941                             break;
942                     }
943                     if (strncmp(arg, "-Wa,", 4) == 0 ||
944                         strncmp(arg, "-Wp,", 4) == 0 ||
945                         strncmp(arg, "-Wl,", 4) == 0) {
946                             newae(ctx->i_ae, arg);
947                             break;
948                     }
1045                    if (strcmp(arg, "-W0,-xc99=pragma") == 0) {
1046                            /* (undocumented) enables _Pragma */
1047                            break;
1048                    }
1049                    if (strcmp(arg, "-W0,-xc99=%none") == 0) {
1050                            /*
```

```
1051                                    * This is a polite way of saying
1052                                    * "no c99 constructs allowed!"
1053                                    * For now, just accept and ignore this.
1054                                    */
1055                                   break;
1056                           }
 949                           if (strcmp(arg, "-W0,-noglobal") == 0 ||
 950                               strcmp(arg, "-W0,-xglobalstatic") == 0) {
 951                                   /*
 952                                    * gcc doesn't prefix local symbols
 953                                    * in debug mode, so this is not needed.
 954                                    */
 955                                   break;
 956                           }
 957                           if (strcmp(arg, "-W0,-Lt") == 0) {
 958                                   /*
 959                                    * Generate tests at the top of loops.
 960                                    * There is no direct gcc equivalent, ignore.
 961                                    */
 962                                   break;
 963                           }
 964                           if (strcmp(arg, "-W0,-xdbggen=no%usedonly") == 0) {
 965                                   newae(ctx->i_ae,
 966                                       "-fno-eliminate-unused-debug-symbols");
 967                                   newae(ctx->i_ae,
 968                                       "-fno-eliminate-unused-debug-types");
 969                                   break;
 970                           }
 971                           if (strcmp(arg, "-W2,-xwrap_int") == 0) {
 972                                   /*
 973                                    * Use the legacy behaviour (pre-SS11)
 974                                    * for integer wrapping.
 975                                    * gcc does not need this.
 976                                    */
 977                                   break;
 978                           }
1087                           if (strcmp(arg, "-W2,-Rcond_elim") == 0) {
1088                                   /*
1089                                    * Elimination and expansion of conditionals;
1090                                    * gcc has no direct equivalent.
1091                                    */
1092                                   break;
1093                           }
 979                           if (strcmp(arg, "-Wd,-xsafe=unboundsym") == 0) {
 980                                   /*
 981                                    * Prevents optimizing away checks for
 982                                    * unbound weak symbol addresses.  gcc does
 983                                    * not do this, so it's not needed.
 984                                    */
 985                                   break;
 986                           }
 987                           if (strncmp(arg, "-Wc,-xcode=", 11) == 0) {
 988                                   xlate(ctx->i_ae, arg + 11, xcode_tbl);
 989                                   if (strncmp(arg + 11, "pic", 3) == 0)
 990                                           pic = 1;
 991                                   break;
 992                           }
 993                           if (strncmp(arg, "-Wc,-Qiselect", 13) == 0) {
 994                                   /*
 995                                    * Prevents insertion of register symbols.
 996                                    * gcc doesn't do this, so ignore it.
 997                                    */
 998                                   break;
 999                           }
1000                           if (strcmp(arg, "-Wc,-Qassembler-ounrefsym=0") == 0) {
1001                                   /*
```



```
1002                                    * Prevents optimizing away of static variables.
1003                                    * gcc does not do this, so it's not needed.
1004                                    */
1005                                   break;
1006                           }
1007 #if defined(__x86)
1123                           if (strcmp(arg, "-Wu,-xmodel=kernel") == 0) {
1124                                   newae(ctx->i_ae, "-ffreestanding");
1125                                   newae(ctx->i_ae, "-mno-red-zone");
1126                                   model = "-mcmodel=kernel";
1127                                   nolibc = 1;
1128                                   break;
1129                           }
1008                           if (strcmp(arg, "-Wu,-save_args") == 0) {
1009                                   newae(ctx->i_ae, "-msave-args");
1010                                   break;
1011                           }
1012 #endif  /* __x86 */
1013                           error(arg);
1014                           break;
1015                   case 'X':
1016                           if (strcmp(arg, "-Xa") == 0 ||
1017                               strcmp(arg, "-Xt") == 0) {
1018                                   Xamode(ctx->i_ae);
1019                                   break;
1020                           }
1143                           if (strcmp(arg, "-Xc") == 0) {
1144                                   Xcmode(ctx->i_ae);
1145                                   break;
1146                           }
1021                           if (strcmp(arg, "-Xs") == 0) {
1022                                   Xsmode(ctx->i_ae);
1023                                   break;
1024                           }
1025                           error(arg);
1026                           break;
1027                   case 'x':
1028                           if (arglen == 1)
1029                                   error(arg);
1030                           switch (arg[2]) {
1157 #if defined(__x86)
1158                           case '3':
1159                                   if (strcmp(arg, "-x386") == 0) {
1160                                           newae(ctx->i_ae, "-march=i386");
1161                                           break;
1162                                   }
1163                                   error(arg);
1164                                   break;
1165                           case '4':
1166                                   if (strcmp(arg, "-x486") == 0) {
1167                                           newae(ctx->i_ae, "-march=i486");
1168                                           break;
1169                                   }
1170                                   error(arg);
1171                                   break;
1172 #endif  /* __x86 */
1031                           case 'a':
1032                                   if (strncmp(arg, "-xarch=", 7) == 0) {
1033                                           mflag |= xlate_xtb(ctx->i_ae, arg + 7);
1034                                           break;
1035                                   }
1036                                   error(arg);
1037                                   break;
1038                           case 'b':
1039                                   if (strncmp(arg, "-xbuiltin=", 10) == 0) {
1040                                           if (strcmp(arg + 10, "%all"))
```

```
1041                                            newae(ctx->i_ae, "-fbuiltin");
1042                                    break;
1043                            }
1044                            error(arg);
1045                            break;
1046                    case 'C':
1047                            /* Accept C++ style comments -- ignore */
1048                            if (strcmp(arg, "-xCC") == 0)
1049                                    break;
1050                            error(arg);
1051                            break;
1052                    case 'c':
1053                            if (strncmp(arg, "-xc99=%all", 10) == 0) {
1054                                    newae(ctx->i_ae, "-std=gnu99");
1055                                    break;
1056                            }
1057                            if (strncmp(arg, "-xc99=%none", 11) == 0) {
1058                                    newae(ctx->i_ae, "-std=gnu89");
1059                                    break;
1060                            }
1061                            if (strncmp(arg, "-xchip=", 7) == 0) {
1062                                    xlate(ctx->i_ae, arg + 7, xchip_tbl);
1063                                    break;
1064                            }
1065                            if (strncmp(arg, "-xcode=", 7) == 0) {
1066                                    xlate(ctx->i_ae, arg + 7, xcode_tbl);
1067                                    if (strncmp(arg + 7, "pic", 3) == 0)
1068                                            pic = 1;
1069                                    break;
1070                            }
1213                            if (strncmp(arg, "-xcache=", 8) == 0)
1214                                    break;
1071                            if (strncmp(arg, "-xcrossfile", 11) == 0)
1072                                    break;
1073                            error(arg);
1074                            break;
1075                    case 'd':
1220                            if (strcmp(arg, "-xdepend") == 0)
1221                                    break;
1076                            if (strncmp(arg, "-xdebugformat=", 14) == 0)
1077                                    break;
1078                            error(arg);
1079                            break;
1080                    case 'F':
1081                            /*
1082                             * Compile for mapfile reordering, or unused
1083                             * section elimination, syntax can be -xF or
1084                             * more complex, like -xF=%all -- ignore.
1085                             */
1086                            if (strncmp(arg, "-xF", 3) == 0)
1087                                    break;
1088                            error(arg);
1089                            break;
1090                    case 'i':
1091                            if (strncmp(arg, "-xinline", 8) == 0)
1092                                    /* No inlining; ignore */
1093                                    break;
1094                            if (strcmp(arg, "-xildon") == 0 ||
1095                                strcmp(arg, "-xildoff") == 0)
1096                                    /* No incremental linking; ignore */
1097                                    break;
1098                            error(arg);
1099                            break;
1100 #if defined(__x86)
1101                    case 'm':
1102                            if (strcmp(arg, "-xmodel=kernel") == 0) {
```



```
1103                                    newae(ctx->i_ae, "-ffreestanding");
1104                                    newae(ctx->i_ae, "-mno-red-zone");
1105                                    model = "-mcmodel=kernel";
1106                                    nolibc = 1;
1107                                    break;
1108                            }
1109                            error(arg);
1110                            break;
1111 #endif  /* __x86 */
1258                    case 'M':
1259                            if (strcmp(arg, "-xM") == 0) {
1260                                    newae(ctx->i_ae, "-M");
1261                                    break;
1262                            }
1263                            if (strcmp(arg, "-xM1") == 0) {
1264                                    newae(ctx->i_ae, "-MM");
1265                                    break;
1266                            }
1267                            error(arg);
1268                            break;
1269                    case 'n':
1270                            if (strcmp(arg, "-xnolib") == 0) {
1271                                    nolibc = 1;
1272                                    break;
1273                            }
1274                            error(arg);
1275                            break;
1112                    case 'O':
1113                            if (strncmp(arg, "-xO", 3) == 0) {
1114                                    size_t len = strlen(arg);
1115                                    char *s = NULL;
1116                                    int c = *(arg + 3);
1117                                    int level;

1119                                    if (len != 4 || !isdigit(c))
1120                                            error(arg);

1122                                    level = atoi(arg + 3);
1123                                    if (level > 5)
1124                                            error(arg);
1125                                    if (level >= 2) {
1126                                            /*
1127                                             * For gcc-3.4.x at -O2 we
1128                                             * need to disable optimizations
1129                                             * that break ON.
1130                                             */
1131                                            optim_disable(ctx->i_ae, level);
1132                                            /*
1133                                             * limit -xO3 to -O2 as well.
1134                                             */
1135                                            level = 2;
1136                                    }
1137                                    if (asprintf(&s, "-O%d", level) == -1)
1138                                            nomem();
1139                                    newae(ctx->i_ae, s);
1140                                    free(s);
1141                                    break;
1142                            }
1143                            error(arg);
1144                            break;
1309                    case 'p':
1310                            if (strcmp(arg, "-xpentium") == 0) {
1311                                    newae(ctx->i_ae, "-march=pentium");
1312                                    break;
1313                            }
1314                            if (strcmp(arg, "-xpg") == 0) {
```

```
1315                                    newae(ctx->i_ae, "-pg");
1316                                    break;
1317                            }
1318                            error(arg);
1319                            break;
1145                    case 'r':
1146                            if (strncmp(arg, "-xregs=", 7) == 0) {
1147                                    xlate(ctx->i_ae, arg + 7, xregs_tbl);
1148                                    break;
1149                            }
1150                            error(arg);
1151                            break;
1152                    case 's':
1153                            if (strcmp(arg, "-xs") == 0 ||
1154                                strcmp(arg, "-xspace") == 0 ||
1155                                strcmp(arg, "-xstrconst") == 0)
1156                                    break;
1157                            error(arg);
1158                            break;
1159                    case 't':
1335                            if (strcmp(arg, "-xtransition") == 0) {
1336                                    newae(ctx->i_ae, "-Wtransition");
1337                                    break;
1338                            }
1339                            if (strcmp(arg, "-xtrigraphs=yes") == 0) {
1340                                    newae(ctx->i_ae, "-trigraphs");
1341                                    break;
1342                            }
1343                            if (strcmp(arg, "-xtrigraphs=no") == 0) {
1344                                    newae(ctx->i_ae, "-notrigraphs");
1345                                    break;
1346                            }
1160                            if (strncmp(arg, "-xtarget=", 9) == 0) {
1161                                    xlate(ctx->i_ae, arg + 9, xtarget_tbl);
1162                                    break;
1163                            }
1164                            error(arg);
1165                            break;
1166                    case 'e':
1167                    case 'h':
1168                    case 'l':
1169                    default:
1170                            error(arg);
1171                            break;
1172                    }
1173                    break;
1174            case 'Y':
1175                    if (arglen == 1) {
1176                            if ((arg = *++ctx->i_oldargv) == NULL ||
1177                                *arg == '\0')
1178                                    error("-Y");
1179                            ctx->i_oldargc--;
1180                            arglen = strlen(arg + 1);
1181                    } else {
1182                            arg += 2;
1183                    }
1184                    /* Just ignore -YS,... for now */
1185                    if (strncmp(arg, "S,", 2) == 0)
1186                            break;
1187                    if (strncmp(arg, "l,", 2) == 0) {
1188                            char *s = strdup(arg);
1189                            s[0] = '-';
1190                            s[1] = 'B';
1191                            newae(ctx->i_ae, s);
1192                            free(s);
1193                            break;
```



```
1194                    }
1195                    if (strncmp(arg, "I,", 2) == 0) {
1196                            char *s = strdup(arg);
1197                            s[0] = '-';
1198                            s[1] = 'I';
1199                            newae(ctx->i_ae, "-nostdinc");
1200                            newae(ctx->i_ae, s);
1201                            free(s);
1202                            break;
1203                    }
1204                    error(arg);
1205                    break;
1206            case 'Q':
1207                    /*
1208                     * We could map -Qy into -Wl,-Qy etc.
1209                     */
1210            default:
1211                    error(arg);
1212                    break;
1213            }
1214    }

1216    free(nameflag);

1218    if (c_files > 1 && (ctx->i_flags & CW_F_SHADOW) &&
1219        op != CW_O_PREPROCESS) {
1220            errx(2, "multiple source files are "
1221                "allowed only with -E or -P");
1222    }

1224    /*
1225     * Make sure that we do not have any unintended interactions between
1226     * the xarch options passed in and the version of the Studio compiler
1227     * used.
1228     */
1229    if ((mflag & (SS11|SS12)) == (SS11|SS12)) {
1230            errx(2,
1231                "Conflicting \"-xarch=\" flags (both Studio 11 and 12)\n");
1232    }

1234    switch (mflag) {
1235    case 0:
1236            /* FALLTHROUGH */
1237    case M32:
1238 #if defined(__sparc)
1239            /*
1240             * Only -m32 is defined and so put in the missing xarch
1241             * translation.
1242             */
1243            newae(ctx->i_ae, "-mcpu=v8");
1244            newae(ctx->i_ae, "-mno-v8plus");
1245 #endif
1246            break;
1247    case M64:
1248 #if defined(__sparc)
1249            /*
1250             * Only -m64 is defined and so put in the missing xarch
1251             * translation.
1252             */
1253            newae(ctx->i_ae, "-mcpu=v9");
1254 #endif
1255            break;
1256    case SS12:
1257 #if defined(__sparc)
1258            /* no -m32/-m64 flag used - this is an error for sparc builds */
1259            (void) fprintf(stderr, "No -m32/-m64 flag defined\n");
```

```
1260                 exit(2);
1261 #endif
1262                 break;
1263         case SS11:
1264                 /* FALLTHROUGH */
1265         case (SS11|M32):
1266         case (SS11|M64):
1267                 break;
1268         case (SS12|M32):
1269 #if defined(__sparc)
1270                 /*
1271                  * Need to add in further 32 bit options because with SS12
1272                  * the xarch=sparcvis option can be applied to 32 or 64
1273                  * bit, and so the translatation table (xtbl) cannot handle
1274                  * that.
1275                  */
1276                 newae(ctx->i_ae, "-mv8plus");
1277 #endif
1278                 break;
1279         case (SS12|M64):
1280                 break;
1281         default:
1282                 (void) fprintf(stderr,
1283                     "Incompatible -xarch= and/or -m32/-m64 options used.\n");
1284                 exit(2);
1285         }

1287         if ((op == CW_O_LINK || op == CW_O_PREPROCESS) &&
1288             (ctx->i_flags & CW_F_SHADOW))
1289                 exit(0);

1291         if (model && !pic)
1292                 newae(ctx->i_ae, model);
1293         if (!nolibc)
1294                 newae(ctx->i_ae, "-lc");
1295         if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1296                 newae(ctx->i_ae, "-o");
1297                 newae(ctx->i_ae, ctx->i_discard);
1298         }
1299 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   58080 Thu Oct  4 22:48:36 2018
new/usr/src/tools/scripts/nightly.sh
9868 unused cw translations should be removed
**********************************************************
______unchanged_portion_omitted_

```

```
 163 #
 164 # Function to do the build, including package generation.
 165 # usage: build LABEL SUFFIX ND MULTIPROTO
 166 # - LABEL is used to tag build output.
 167 # - SUFFIX is used to distinguish files (e.g., DEBUG vs non-DEBUG,
 168 #   open-only vs full tree).
 169 # - ND is "-nd" (non-DEBUG builds) or "" (DEBUG builds).
 170 # - If MULTIPROTO is "yes", it means to name the proto area according to
 171 #   SUFFIX.  Otherwise ("no"), (re)use the standard proto area.
 172 #
 173 function build {
 174         LABEL=$1
 175         SUFFIX=$2
 176         ND=$3
 177         MULTIPROTO=$4
 178         INSTALLOG=install${SUFFIX}-${MACH}
 179         NOISE=noise${SUFFIX}-${MACH}
 180         PKGARCHIVE=${PKGARCHIVE_ORIG}${SUFFIX}

 182         ORIGROOT=$ROOT
 183         [ $MULTIPROTO = no ] || export ROOT=$ROOT$SUFFIX

 185         export ENVLDLIBS1='myldlibs $ROOT'
 186         export ENVCPPFLAGS1='myheaders $ROOT'

 188         this_build_ok=y
 189         #
 190         #       Build OS-Networking source
 191         #
 192         echo "\n==== Building OS-Net source at 'date' ($LABEL) ====\n" \
 193                 >> $LOGFILE

 195         rm -f $SRC/${INSTALLOG}.out
 196         cd $SRC
 197         /bin/time $MAKE -e install 2>&1 | \
 198             tee -a $SRC/${INSTALLOG}.out >> $LOGFILE

 200         echo "\n==== Build errors ($LABEL) ====\n" >> $mail_msg_file
 201         egrep ":" $SRC/${INSTALLOG}.out |
 202             egrep -e "(^${MAKE}:|[      ]error[:        \n])" | \
 203             egrep -v "Ignoring unknown host" | \
 204             egrep -v "cc .* -o error " | \
 205             egrep -v "warning" | tee $TMPDIR/build_errs${SUFFIX} \
 206             >> $mail_msg_file
 207             sed -n "/^Undefined[        ]*first referenced$/,/^ld: fatal:/p" \
 208             < $SRC/${INSTALLOG}.out >> $mail_msg_file
 209         if [[ -s $TMPDIR/build_errs${SUFFIX} ]]; then
 210                 build_ok=n
 211                 this_build_ok=n
 212         fi
 213         grep "bootblock image is .* bytes too big" $SRC/${INSTALLOG}.out \
 214                 >> $mail_msg_file
 215         if [ "$?" = "0" ]; then
 216                 build_ok=n
 217                 this_build_ok=n
 218         fi

 220         echo "\n==== Build warnings ($LABEL) ====\n" >>$mail_msg_file
 221         egrep -i warning: $SRC/${INSTALLOG}.out \
```



```
 222                 | egrep -v '^tic:' \
 223                 | egrep -v "symbol (\'|')timezone' has differing types:" \
 224                 | egrep -v "parameter <PSTAMP> set to" \
 225                 | egrep -v "Ignoring unknown host" \
 226                 | egrep -v "redefining segment flags attribute for" \
 227                 | tee $TMPDIR/build_warnings${SUFFIX} >> $mail_msg_file
 228         if [[ -s $TMPDIR/build_warnings${SUFFIX} ]]; then
 229                 build_ok=n
 230                 this_build_ok=n
 231         fi

 233         echo "\n==== Ended OS-Net source build at 'date' ($LABEL) ====\n" \
 234                 >> $LOGFILE

 236         echo "\n==== Elapsed build time ($LABEL) ====\n" >>$mail_msg_file
 237         tail -3  $SRC/${INSTALLOG}.out >>$mail_msg_file

 239         if [ "$i_FLAG" = "n" ]; then
 240                 rm -f $SRC/${NOISE}.ref
 241                 if [ -f $SRC/${NOISE}.out ]; then
 242                         mv $SRC/${NOISE}.out $SRC/${NOISE}.ref
 243                 fi
 244                 grep : $SRC/${INSTALLOG}.out \
 245                         | egrep -v '^/' \
 246                         | egrep -v '^(Start|Finish|real|user|sys|./bld_awk)' \
 247                         | egrep -v '^tic:' \
 248                         | egrep -v '^mcs' \
 249                         | egrep -v '^LD_LIBRARY_PATH=' \
 250                         | egrep -v 'ar: creating' \
 251                         | egrep -v 'ar: writing' \
 252                         | egrep -v 'conflicts:' \
 253                         | egrep -v ':saved created' \
 254                         | egrep -v '^stty.*c:' \
 255                         | egrep -v '^mfgname.c:' \
 256                         | egrep -v '^uname-i.c:' \
 257                         | egrep -v '^volumes.c:' \
 258                         | egrep -v '^lint library construction:' \
 259                         | egrep -v 'tsort: INFORM:' \
 260                         | egrep -v 'stripalign:' \
 261                         | egrep -v 'chars, width' \
 262                         | egrep -v "symbol (\'|')timezone' has differing types:"
 263                         | egrep -v 'PSTAMP' \
 264                         | egrep -v '^Manifying' \
 265                         | egrep -v 'Ignoring unknown host' \
 266                         | egrep -v 'Processing method:' \
 267                         | egrep -v '^Writing' \
 268                         | egrep -v 'spellin1:' \
 269                         | egrep -v '^adding:' \
 270                         | egrep -v "^echo 'msgid" \
 271                         | egrep -v '^echo ' \
 272                         | egrep -v '\.c:$' \
 273                         | egrep -v '^Adding file:' \
 274                         | egrep -v 'CLASSPATH=' \
 275                         | egrep -v '\/var\/mail\/:saved' \
 276                         | egrep -v -- '-DUTS_VERSION=' \
 277                         | egrep -v '^Running Mkbootstrap' \
 278                         | egrep -v '^Applet length read:' \
 279                         | egrep -v 'bytes written:' \
 280                         | egrep -v '^File:SolarisAuthApplet.bin' \
 281                         | egrep -v -i 'jibversion' \
 282                         | egrep -v '^Output size:' \
 283                         | egrep -v '^Solo size statistics:' \
 284                         | egrep -v '^Using ROM API Version' \
 285                         | egrep -v '^Zero Signature length:' \
 286                         | egrep -v '^Note \(probably harmless\):' \
 287                         | egrep -v '::' \
```

```
288                         | egrep -v -- '-xcache' \
288                         | egrep -v '^\+' \
290                         | egrep -v '^cc1: note: -fwritable-strings' \
289                         | egrep -v 'svccfg-native -s svc:/' \
290                         | sort | uniq >$SRC/${NOISE}.out
291                 if [ ! -f $SRC/${NOISE}.ref ]; then
292                         cp $SRC/${NOISE}.out $SRC/${NOISE}.ref
293                 fi
294                 echo "\n==== Build noise differences ($LABEL) ====\n" \
295                         >>$mail_msg_file
296                 diff $SRC/${NOISE}.ref $SRC/${NOISE}.out >>$mail_msg_file
297         fi

299         #
300         #       Re-sign selected binaries using signing server
301         #       (gatekeeper builds only)
302         #
303         if [ -n "$CODESIGN_USER" -a "$this_build_ok" = "y" ]; then
304                 echo "\n==== Signing proto area at 'date' ====\n" >> $LOGFILE
305                 signing_file="${TMPDIR}/signing"
306                 rm -f ${signing_file}
307                 export CODESIGN_USER
308                 signproto $SRC/tools/codesign/creds 2>&1 | \
309                         tee -a ${signing_file} >> $LOGFILE
310                 echo "\n==== Finished signing proto area at 'date' ====\n" \
311                     >> $LOGFILE
312                 echo "\n==== Crypto module signing errors ($LABEL) ====\n" \
313                     >> $mail_msg_file
314                 egrep 'WARNING|ERROR' ${signing_file} >> $mail_msg_file
315                 if (( $? == 0 )) ; then
316                         build_ok=n
317                         this_build_ok=n
318                 fi
319         fi

321         #
322         #       Building Packages
323         #
324         if [ "$p_FLAG" = "y" -a "$this_build_ok" = "y" ]; then
325                 if [ -d $SRC/pkg ]; then
326                         echo "\n==== Creating $LABEL packages at 'date' ====\n"
327                                 >> $LOGFILE
328                         echo "Clearing out $PKGARCHIVE ..." >> $LOGFILE
329                         rm -rf $PKGARCHIVE >> "$LOGFILE" 2>&1
330                         mkdir -p $PKGARCHIVE >> "$LOGFILE" 2>&1

332                         rm -f $SRC/pkg/${INSTALLOG}.out
333                         cd $SRC/pkg
334                         /bin/time $MAKE -e install 2>&1 | \
335                             tee -a $SRC/pkg/${INSTALLOG}.out >> $LOGFILE

337                         echo "\n==== package build errors ($LABEL) ====\n" \
338                                 >> $mail_msg_file

340                         egrep "${MAKE}|ERROR|WARNING" $SRC/pkg/${INSTALLOG}.out
341                                 grep ':' | \
342                                 grep -v PSTAMP | \
343                                 egrep -v "Ignoring unknown host" | \
344                                 tee $TMPDIR/package >> $mail_msg_file
345                         if [[ -s $TMPDIR/package ]]; then
346                                 build_extras_ok=n
347                                 this_build_ok=n
348                         fi
349                 else
350                         #
351                         # Handle it gracefully if -p was set but there so
```



```
352                         # no pkg directory.
353                         #
354                         echo "\n==== No $LABEL packages to build ====\n" \
355                                 >> $LOGFILE
356                 fi
357         else
358                 echo "\n==== Not creating $LABEL packages ====\n" >> $LOGFILE
359         fi

361         ROOT=$ORIGROOT
362 }
```

*unchanged_portion_omitted_*