```
**********************************************************
   35973 Thu Jan 25 19:54:33 2018
new/usr/src/Makefile.master
8571 Makefile.master should not trust $PATH
**********************************************************
```

```
   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 #

  32 #
  33 # Makefile.master, global definitions for system source
  34 #
  35 ROOT=           /proto

  37 #
  38 # Adjunct root, containing an additional proto area to be used for headers
  39 # and libraries.
  40 #
  41 ADJUNCT_PROTO=

  43 #
  44 # Adjunct for building things that run on the build machine.
  45 #
  46 NATIVE_ADJUNCT= /usr

  48 #
  49 # RELEASE_BUILD should be cleared for final release builds.
  50 # NOT_RELEASE_BUILD is exactly what the name implies.
  51 #
  52 # __GNUC toggles the building of ON components using gcc and related tools.
  53 # Normally set to '#', set it to '' to do gcc build.
  54 #
  55 # The declaration POUND_SIGN is always '#'. This is needed to get around the
  56 # make feature that '#' is always a comment delimiter, even when escaped or
  57 # quoted. We use this macro expansion method to get POUND_SIGN rather than
  58 # always breaking out a shell because the general case can cause a noticable
  59 # slowdown in build times when so many Makefiles include Makefile.master.
  60 #
  61 # While the majority of users are expected to override the setting below
```



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

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

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

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

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

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

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

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

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

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

 118 # define runtime JAVA_HOME, primarily for cmd/pools/poold
 119 JAVA_HOME=      /usr/java
 120 # define buildtime JAVA_ROOT
 121 JAVA_ROOT=      /usr/java
 122 # define to build with JAVA 8
 123 BLD_JAVA_8=     $(POUND_SIGN)

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

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

131 RPCGEN=         /usr/bin/rpcgen
132 STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
133 ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
134 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
135 BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
136 VTFONTCVT=      $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcvt
137 # echo(1) and true(1) are specified without absolute paths, so that the shell
138 # spawned by make(1) may use the built-in versions.  This is minimally
139 # problematic, as the shell spawned by make(1) is known and under control, the
140 # only risk being if the shell falls back to $PATH.
141 #
142 # We specifically want an echo(1) that does interpolation of escape sequences,
143 # which ksh93, /bin/sh, and bash will all provide.
144 #endif /* ! codereview */
145 ECHO=           echo
137 INS=            install
146 TRUE=           true
147 INS=            $(ONBLD_TOOLS)/bin/$(MACH)/install
148 #endif /* ! codereview */
149 SYMLINK=        /usr/bin/ln -s
150 LN=             /usr/bin/ln
151 CHMOD=          /usr/bin/chmod
152 MV=             /usr/bin/mv -f
153 RM=             /usr/bin/rm -f
154 CUT=            /usr/bin/cut
155 NM=             /usr/ccs/bin/nm
156 DIFF=           /usr/bin/diff
157 GREP=           /usr/bin/grep
158 EGREP=          /usr/bin/egrep
159 ELFWRAP=        /usr/bin/elfwrap
160 KSH93=          /usr/bin/ksh93
161 SED=            /usr/bin/sed
162 AWK=            /usr/bin/nawk
163 CP=             /usr/bin/cp -f
164 MCS=            /usr/ccs/bin/mcs
165 CAT=            /usr/bin/cat
166 ELFDUMP=        /usr/ccs/bin/elfdump
167 M4=             /usr/bin/m4
168 STRIP=          /usr/ccs/bin/strip
169 LEX=            /usr/ccs/bin/lex
170 FLEX=           /usr/bin/flex
171 YACC=           /usr/ccs/bin/yacc
172 CPP=            /usr/lib/cpp
173 JAVAC=          $(JAVA_ROOT)/bin/javac
174 JAVAH=          $(JAVA_ROOT)/bin/javah
175 JAVADOC=        $(JAVA_ROOT)/bin/javadoc
176 RMIC=           $(JAVA_ROOT)/bin/rmic
177 JAR=            $(JAVA_ROOT)/bin/jar
178 CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
179 CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
180 CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
181 CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
182 NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
183 GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
184 XREF=           $(ONBLD_TOOLS)/bin/xref
185 FIND=           /usr/bin/find
186 PERL=           /usr/bin/perl
187 PERL_VERSION=   5.10.0
188 PERL_PKGVERS=   -510
189 PERL_ARCH =            i86pc-solaris-64int
190 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
191 PYTHON_VERSION= 2.7
192 PYTHON_PKGVERS= -27
```



```
193 PYTHON=         /usr/bin/python$(PYTHON_VERSION)
194 SORT=           /usr/bin/sort
195 TOUCH=          /usr/bin/touch
196 WC=             /usr/bin/wc
197 XARGS=          /usr/bin/xargs
198 ELFEDIT=        /usr/bin/elfedit
199 DTRACE=         /usr/sbin/dtrace -xnolibs
200 UNIQ=           /usr/bin/uniq
201 TAR=            /usr/bin/tar
202 ASTBINDIR=      /usr/ast/bin
203 MSGCC=          $(ASTBINDIR)/msgcc
204 MSGFMT=         /usr/bin/msgfmt -s
205 LCDEF=          $(ONBLD_TOOLS)/bin/$(MACH)/localedef
206 TIC=            $(ONBLD_TOOLS)/bin/$(MACH)/tic
207 ZIC=            $(ONBLD_TOOLS)/bin/$(MACH)/zic

209 FILEMODE=       644
210 DIRMODE=        755

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

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

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

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

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

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

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

251 # install a link
252 INSLINKTARGET=  $<
253 INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@
254 INS.symlink=    $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

256 #
257 # Python bakes the mtime of the .py file into the compiled .pyc and
258 # rebuilds if the baked-in mtime != the mtime of the source file
```

```
# (rather than only if it's less than), thus when installing python
# files we must make certain to not adjust the mtime of the source
# (.py) file.
#
INS.pyfile=     $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $

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

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

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

sparc_BUILD64=
i386_BUILD64=
BUILD64=        $($(MACH)_BUILD64)

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

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

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

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

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



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

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

#
# compiler '-xarch' flag. This is here to centralize it and make it
# overridable for testing.
sparc_XARCH=    -m32
sparcv9_XARCH=  -m64
i386_XARCH=     -m32
amd64_XARCH=    -m64 -Ui386 -U__i386

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

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

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

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

#
# disable the incremental linker
ILDOFF=                 -xildoff
#
XDEPEND=                -xdepend
XFFLAG=                 -xF=%all
```

```
391 XESS=                   -xs
392 XSTRCONST=              -xstrconst

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

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

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

409 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
410 # -nd builds
411 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
412 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

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

418 # C99 mode
419 C99_ENABLE=     -xc99=%all
420 C99_DISABLE=    -xc99=%none
421 C99MODE=        $(C99_DISABLE)
422 C99LMODE=       $(C99MODE:-xc99%=-Xc99%)

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

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

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

443 COPTFLAG= $($(MACH)_COPTFLAG)
444 COPTFLAG64= $($(MACH64)_COPTFLAG)

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

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



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

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

468 #
469 # Flags used to build in debug mode for ctf generation.  Bugs in the Devpro
470 # compilers currently prevent us from building with cc-emitted DWARF.
471 #
472 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
473 CTF_FLAGS_i386  = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)

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

478 # Sun Studio produces broken userland code when saving arguments.
479 $(__GNUC)CTF_FLAGS_amd64 += $(SAVEARGS)

481 CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
482 CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
483 CTF_FLAGS       = $(CTF_FLAGS_32)

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

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

495 OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
496         $(CC) $(GOFLAGS) $(CFLAGS64) $(CPPFLAGS)

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

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

509 #
510 # The Sun Studio 11 compiler has changed the behaviour of integer
511 # wrap arounds and so a flag is needed to use the legacy behaviour
512 # (without this flag panics/hangs could be exposed within the source).
513 #
514 sparc_IROPTFLAG         = -W2,-xwrap_int
515 sparcv9_IROPTFLAG       = -W2,-xwrap_int
516 i386_IROPTFLAG          =
517 amd64_IROPTFLAG         =

519 IROPTFLAG               = $($(MACH)_IROPTFLAG)
520 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)

522 sparc_XREGSFLAG         = -xregs=no%appl
```

```
sparcv9_XREGSFLAG       = -xregs=no%appl
i386_XREGSFLAG          =
amd64_XREGSFLAG         =

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

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

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

CSOURCEDEBUGFLAGS       =
CCSOURCEDEBUGFLAGS      =
$(SRCDBGBLD)CSOURCEDEBUGFLAGS   = -g -xs
$(SRCDBGBLD)CCSOURCEDEBUGFLAGS  = -g -xs

CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
                $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
                $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
                $(CUSERFLAGS)
CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
                $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
                $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
                $(CUSERFLAGS64)
#
# Flags that are used to build parts of the code that are subsequently
# run on the build machine (also known as the NATIVE_BUILD).
#
NATIVE_CFLAGS=  $(COPTFLAG) $($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
                $(ILDOFF) $(CERRWARN) $(C99MODE) $($(NATIVE_MACH)_CCUNBOUND) \
                $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
                $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

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

#
# For source message catalogue
#
.SUFFIXES: $(SUFFIXES) .i .po
MSGROOT= $(ROOT)/catalog
MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
```



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

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

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

sparc_CCFLAGS=          -cg92 -compat=4 \
                        -Qoption ccfe -messages=no%anachronism \
                        $(CCERRWARN)
sparcv9_CCFLAGS=        $(sparcv9_XARCH) -dalign -compat=5 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCCREGSYM) \
                        $(CCERRWARN)
i386_CCFLAGS=           -compat=4 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCERRWARN)
amd64_CCFLAGS=          $(amd64_XARCH) -compat=5 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCERRWARN)

sparc_CCOPTFLAG=        -O
sparcv9_CCOPTFLAG=      -O
i386_CCOPTFLAG=         -O
amd64_CCOPTFLAG=        -O

CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
                $(CCUSERFLAGS)
CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
                $(CCUSERFLAGS64)

#
#
#
ELFWRAP_FLAGS   =
ELFWRAP_FLAGS64 =       -64

#
# Various mapfiles that are used throughout the build, and delivered to
# /usr/lib/ld.
#
MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
MAPFILE.NED_sparc =
MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
```

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

657 #
658 # Generated mapfiles that are compiler specific, and used throughout the
659 # build.  These mapfiles are not delivered in /usr/lib/ld.
660 #
661 MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
662 $(__GNUC64)MAPFILE.NGB_sparc= \
663                         $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
664 MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
665 $(__GNUC64)MAPFILE.NGB_sparcv9= \
666                         $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
667 MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
668 $(__GNUC64)MAPFILE.NGB_i386= \
669                         $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
670 MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
671 $(__GNUC64)MAPFILE.NGB_amd64= \
672                         $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
673 MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

675 #
676 # A generic interface mapfile name, used by various dynamic objects to define
677 # the interfaces and interposers the object must export.
678 #
679 MAPFILE.INT =           mapfile-intf

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

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

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

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

717 #
718 # Link time macros
719 #
720 CCNEEDED                = -lC
```



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

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

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

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

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

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

775 # Define native compilation macros
776 #

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

785 # Till SS12u1 formally becomes the NV CBE, LINT is hard
786 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
```

```
# location. Impacted variables are sparc_LINT, sparcv9_LINT,
# i386_LINT, amd64_LINT.
# Reset them when SS12u1 is rolled out.
#

# Specify platform compiler versions for languages
# that we use (currently only c and c++).
#
sparc_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
$(__GNUC)sparc_CC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
sparc_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
$(__GNUC)sparc_CCC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
sparc_CPP=              /usr/ccs/lib/cpp
sparc_AS=               /usr/ccs/bin/as -xregsym=no
sparc_LD=               /usr/ccs/bin/ld
sparc_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

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

i386_CC=                $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
$(__GNUC)i386_CC=       $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
i386_CCC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
$(__GNUC)i386_CCC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
i386_CPP=               /usr/ccs/lib/cpp
i386_AS=                /usr/ccs/bin/as
$(__GNUC)i386_AS=       $(ONBLD_TOOLS)/bin/$(MACH)/aw
i386_LD=                /usr/ccs/bin/ld
i386_LINT=              $(SPRO_ROOT)/sunstudio12.1/bin/lint

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

NATIVECC=               $($(NATIVE_MACH)_CC)
NATIVECCC=              $($(NATIVE_MACH)_CCC)
NATIVECPP=              $($(NATIVE_MACH)_CPP)
NATIVEAS=               $($(NATIVE_MACH)_AS)
NATIVELD=               $($(NATIVE_MACH)_LD)
NATIVELINT=             $($(NATIVE_MACH)_LINT)

#
# Makefile.master.64 overrides these settings
#
CC=                     $(NATIVECC)
CCC=                    $(NATIVECCC)
CPP=                    $(NATIVECPP)
AS=                     $(NATIVEAS)
LD=                     $(NATIVELD)
LINT=                   $(NATIVELINT)

# The real compilers used for this build
CW_CC_CMD=              $(CC) -_compiler
CW_CCC_CMD=             $(CCC) -_compiler
REAL_CC=                $(CW_CC_CMD:sh)
```



```
REAL_CCC=               $(CW_CCC_CMD:sh)

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

BDIRECT=        -Bdirect
BDYNAMIC=       -Bdynamic
BLOCAL=         -Blocal
BNODIRECT=      -Bnodirect
BREDUCE=        -Breduce
BSTATIC=        -Bstatic

ZDEFS=          -zdefs
ZDIRECT=        -zdirect
ZIGNORE=        -zignore
ZINITFIRST=     -zinitfirst
ZINTERPOSE=     -zinterpose
ZLAZYLOAD=      -zlazyload
ZLOADFLTR=      -zloadfltr
ZMULDEFS=       -zmuldefs
ZNODEFAULTLIB=  -znodefaultlib
ZNODEFS=        -znodefs
ZNODELETE=      -znodelete
ZNODLOPEN=      -znodlopen
ZNODUMP=        -znodump
ZNOLAZYLOAD=    -znolazyload
ZNOLDYNSYM=     -znoldynsym
ZNORELOC=       -znoreloc
ZNOVERSION=     -znoversion
ZRECORD=        -zrecord
ZREDLOCSYM=     -zredlocsym
ZTEXT=          -ztext
ZVERBOSE=       -zverbose

GSHARED=        -G
CCMT=           -mt

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

sparc_C_PICFLAGS =      -K pic
sparcv9_C_PICFLAGS =    -K pic
i386_C_PICFLAGS =       -K pic
amd64_C_PICFLAGS =      -K pic
C_PICFLAGS =            $($(MACH)_C_PICFLAGS)
C_PICFLAGS64 =          $($(MACH64)_C_PICFLAGS)

sparc_C_BIGPICFLAGS =   -K PIC
sparcv9_C_BIGPICFLAGS = -K PIC
i386_C_BIGPICFLAGS =    -K PIC
amd64_C_BIGPICFLAGS =   -K PIC
C_BIGPICFLAGS =         $($(MACH)_C_BIGPICFLAGS)
C_BIGPICFLAGS64 =       $($(MACH64)_C_BIGPICFLAGS)

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

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

#
```

```
# Default label for CTF sections
#
CTFCVTFLAGS=            -i -L VERSION

#
# Override to pass module-specific flags to ctfmerge.  Currently used only by
# 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) $@ $%
```

```
1051         $(PROCESS_COMMENT) $%
1052         $(RM) $%
1053 .y:
1054         $(YACC.y) $<
1055         $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1056         $(POST_PROCESS)
1057         $(RM) y.tab.c
1058 .y.o:
1059         $(YACC.y) $<
1060         $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1061         $(POST_PROCESS_O)
1062         $(RM) y.tab.c
1063 .l:
1064         $(RM) $*.c
1065         $(LEX.l) $< > $*.c
1066         $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1067         $(POST_PROCESS)
1068         $(RM) $*.c
1069 .l.o:
1070         $(RM) $*.c
1071         $(LEX.l) $< > $*.c
1072         $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1073         $(POST_PROCESS_O)
1074         $(RM) $*.c

1076 .bin.o:
1077         $(COMPILE.b) -o $@ $<
1078         $(POST_PROCESS_O)

1080 .java.class:
1081         $(COMPILE.java) $<

1083 # Bourne and Korn shell script message catalog build rules.
1084 # We extract all gettext strings with sed(1) (being careful to permit
1085 # multiple gettext strings on the same line), weed out the dups, and
1086 # build the catalogue with awk(1).

1088 .sh.po .ksh.po:
1089         $(SED) -n -e ":a"                                       \
1090                   -e "h"                                        \
1091                   -e "s/.*gettext *\(\"[^\"]*\"\).*/\1/p"       \
1092                   -e "x"                                        \
1093                   -e "s/\(.*\)gettext *\"[^\"]*\"\(.*\)/\1\2/"  \
1094                   -e "t a"                                      \
1095                $< | sort -u | $(AWK) '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1097 #
1098 # Python and Perl executable and message catalog build rules.
1099 #
1100 .SUFFIXES: .pl .pm .py .pyc

1102 .pl:
1103         $(RM) $@;
1104         $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1105         $(CHMOD) +x $@

1107 .py:
1108         $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $(CHMOD)

1110 .py.pyc:
1111         $(RM) $@
1112         $(PYTHON) -mpy_compile $<
1113         @[ $(<)c = $@ ] || $(MV) $(<)c $@

1115 .py.po:
1116         $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;
```



```
1118 .pl.po .pm.po:
1119         $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1120         $(RM)   $@ ;
1121         $(SED) "/^domain/d" < $(<F).po > $@ ;
1122         $(RM) $(<F).po

1124 #
1125 # When using xgettext, we want messages to go to the default domain,
1126 # rather than the specified one.  This special version of the
1127 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1128 # causing xgettext to put all messages into the default domain.
1129 #
1130 CPPFORPO=$(COMPILE.cpp:\"$(TEXT_DOMAIN)\"=TEXT_DOMAIN)

1132 .c.i:
1133         $(CPPFORPO) $< > $@

1135 .h.i:
1136         $(CPPFORPO) $< > $@

1138 .y.i:
1139         $(YACC) -d $<
1140         $(CPPFORPO) y.tab.c  > $@
1141         $(RM) y.tab.c

1143 .l.i:
1144         $(LEX) $<
1145         $(CPPFORPO) lex.yy.c  > $@
1146         $(RM) lex.yy.c

1148 .c.po:
1149         $(CPPFORPO) $< > $<.i
1150         $(BUILD.po)

1152 .cc.po:
1153         $(CPPFORPO) $< > $<.i
1154         $(BUILD.po)

1156 .y.po:
1157         $(YACC) -d $<
1158         $(CPPFORPO) y.tab.c  > $<.i
1159         $(BUILD.po)
1160         $(RM) y.tab.c

1162 .l.po:
1163         $(LEX) $<
1164         $(CPPFORPO) lex.yy.c  > $<.i
1165         $(BUILD.po)
1166         $(RM) lex.yy.c

1168 #
1169 # Rules to perform stylistic checks
1170 #
1171 .SUFFIXES: .x .xml .check .xmlchk

1173 .h.check:
1174         $(DOT_H_CHECK)

1176 .x.check:
1177         $(DOT_X_CHECK)

1179 .xml.xmlchk:
1180         $(MANIFEST_CHECK)

1182 #
```

```
1183 # Include rules to render automated sccs get rules "safe".
1184 #
1185 include $(SRC)/Makefile.noget
```

```
**********************************************************
   12202 Thu Jan 25 19:54:34 2018
new/usr/src/pkg/manifests/developer-build-onbld.mf
8998 depcheck is useless and should be removed
8992 checkproto is useless and can be deleted
8991 pmodes is useless and can be deleted
8990 /opt/onbld/gk is useless
**********************************************************
```

```
#
# 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) 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2015, Richard Lowe
# Copyright 2012, Piotr Jasiukajtis
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright (c) 2014, Joyent, Inc.
# Copyright 2016 Toomas Soome <tsoome@me.com>
#

set name=pkg.fmri value=pkg:/developer/build/onbld@$(PKGVERS)
set name=pkg.description value="tools used to build the OS-Net consolidation"
set name=pkg.summary value="OS-Net Build Tools"
set name=info.classification \
    value="org.opensolaris.category.2008:Development/Distribution Tools"

#
# This package should not be incorporated.  This allows the tools
# to be upgraded without upgrading the entire system.
#
set name=org.opensolaris.noincorp value=true
set name=variant.arch value=$(ARCH)
dir path=opt group=sys
dir path=opt/onbld
dir path=opt/onbld/bin
dir path=opt/onbld/bin/$(ARCH)
dir path=opt/onbld/env
dir path=opt/onbld/etc
dir path=opt/onbld/etc/exception_lists
dir path=opt/onbld/gk
dir path=opt/onbld/lib
dir path=opt/onbld/lib/$(ARCH)
dir path=opt/onbld/lib/$(ARCH)/64
dir path=opt/onbld/lib/perl
dir path=opt/onbld/lib/python$(PYTHON_VERSION)
dir path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld
dir path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks
dir path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm
```



```
dir path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/hgext
dir path=opt/onbld/man
dir path=opt/onbld/man/man1onbld
dir path=opt/onbld/share
$(i386_ONLY)file path=opt/onbld/bin/$(ARCH)/aw mode=0555
$(i386_ONLY)file path=opt/onbld/bin/$(ARCH)/btxld mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/chk4ubin mode=0555
file path=opt/onbld/bin/$(ARCH)/codereview mode=0555
file path=opt/onbld/bin/$(ARCH)/cscope-fast mode=0555
file path=opt/onbld/bin/$(ARCH)/ctfconvert mode=0555
file path=opt/onbld/bin/$(ARCH)/ctfdump mode=0555
file path=opt/onbld/bin/$(ARCH)/ctfmerge mode=0555
file path=opt/onbld/bin/$(ARCH)/ctfstabs mode=0555
file path=opt/onbld/bin/$(ARCH)/ctfstrip mode=0555
file path=opt/onbld/bin/$(ARCH)/cw mode=0555
$(i386_ONLY)file path=opt/onbld/bin/$(ARCH)/elfextract mode=0555
file path=opt/onbld/bin/$(ARCH)/findunref mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/forth mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/forth_preload.so.1 mode=0555
file path=opt/onbld/bin/$(ARCH)/install mode=0555
file path=opt/onbld/bin/$(ARCH)/lintdump mode=0555
file path=opt/onbld/bin/$(ARCH)/localedef mode=0555
file path=opt/onbld/bin/$(ARCH)/make mode=0555
$(i386_ONLY)file path=opt/onbld/bin/$(ARCH)/mbh_patch mode=0555
file path=opt/onbld/bin/$(ARCH)/ndrgen mode=0555
file path=opt/onbld/bin/$(ARCH)/ndrgen1 mode=0555
file path=opt/onbld/bin/$(ARCH)/pmodes mode=0555
file path=opt/onbld/bin/$(ARCH)/protocmp mode=0555
file path=opt/onbld/bin/$(ARCH)/protolist mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/stabs mode=0555
file path=opt/onbld/bin/$(ARCH)/tic mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/tokenize mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/tokenize.exe mode=0555
file path=opt/onbld/bin/$(ARCH)/zic mode=0555
file path=opt/onbld/bin/Install mode=0555
file path=opt/onbld/bin/bldenv mode=0555
file path=opt/onbld/bin/bringovercheck mode=0555
file path=opt/onbld/bin/build_cscope mode=0555
file path=opt/onbld/bin/cddlchk mode=0555
file path=opt/onbld/bin/check_rtime mode=0555
file path=opt/onbld/bin/checkpaths mode=0555
file path=opt/onbld/bin/checkproto mode=0555
file path=opt/onbld/bin/copyrightchk mode=0555
file path=opt/onbld/bin/cstyle mode=0555
file path=opt/onbld/bin/elfcmp mode=0555
file path=opt/onbld/bin/find_elf mode=0555
file path=opt/onbld/bin/findcrypto mode=0555
file path=opt/onbld/bin/flg.flp mode=0555
file path=opt/onbld/bin/genoffsets mode=0555
file path=opt/onbld/bin/get_depend_info mode=0555
file path=opt/onbld/bin/git-pbchk mode=0555
file path=opt/onbld/bin/hdrchk mode=0555
file path=opt/onbld/bin/hg-active mode=0555
file path=opt/onbld/bin/hgsetup mode=0555
file path=opt/onbld/bin/interface_check mode=0555
file path=opt/onbld/bin/interface_cmp mode=0555
file path=opt/onbld/bin/jstyle mode=0555
file path=opt/onbld/bin/make_pkg_db mode=0555
file path=opt/onbld/bin/mapfilechk mode=0555
file path=opt/onbld/bin/nightly mode=0555
file path=opt/onbld/bin/onu mode=0555
file path=opt/onbld/bin/protocmp.terse mode=0555
file path=opt/onbld/bin/sccscheck mode=0555
file path=opt/onbld/bin/signit mode=0555
file path=opt/onbld/bin/signproto mode=0555
file path=opt/onbld/bin/validate_flg mode=0555
```

```
120 file path=opt/onbld/bin/validate_paths mode=0555
121 file path=opt/onbld/bin/validate_pkg mode=0555
122 file path=opt/onbld/bin/wdiff mode=0555
123 file path=opt/onbld/bin/webrev mode=0555
124 file path=opt/onbld/bin/which_scm mode=0555
125 file path=opt/onbld/bin/ws mode=0555
126 file path=opt/onbld/bin/wsdiff mode=0555
127 file path=opt/onbld/bin/xref mode=0555
128 file path=opt/onbld/bin/xref.mk
129 file path=opt/onbld/env/illumos
135 file path=opt/onbld/etc/SampleLinks
136 file path=opt/onbld/etc/SamplePkgLinks
130 file path=opt/onbld/etc/exception_lists/check_rtime
131 file path=opt/onbld/etc/exception_lists/interface_check
132 file path=opt/onbld/etc/exception_lists/interface_cmp
133 file path=opt/onbld/etc/hgstyle
134 file path=opt/onbld/etc/its.conf
135 file path=opt/onbld/etc/its.reg
143 file path=opt/onbld/gk/.cshrc
144 file path=opt/onbld/gk/.login
145 file path=opt/onbld/gk/gen_make.machines mode=0755
136 file path=opt/onbld/lib/$(ARCH)/64/libmakestate.so.1
137 file path=opt/onbld/lib/$(ARCH)/libdwarf.so.1
138 file path=opt/onbld/lib/$(ARCH)/libmakestate.so.1
139 file path=opt/onbld/lib/perl/onbld_elfmod.pm
140 file path=opt/onbld/lib/perl/onbld_elfmod_vertype.pm
141 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/CStyle.py \
142     mode=0444
143 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/CStyle.pyc \
144     mode=0444
145 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Cddl.py mode=0444
146 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Cddl.pyc \
147     mode=0444
148 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/CmtBlk.py \
149     mode=0444
150 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/CmtBlk.pyc \
151     mode=0444
152 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Comments.py \
153     mode=0444
154 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Comments.pyc \
155     mode=0444
156 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Copyright.py \
157     mode=0444
158 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Copyright.pyc \
159     mode=0444
160 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/DbLookups.py \
161     mode=0444
162 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/DbLookups.pyc \
163     mode=0444
164 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/HdrChk.py \
165     mode=0444
166 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/HdrChk.pyc \
167     mode=0444
168 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/JStyle.py \
169     mode=0444
170 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/JStyle.pyc \
171     mode=0444
172 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Keywords.py \
173     mode=0444
174 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Keywords.pyc \
175     mode=0444
176 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/ManLint.py \
177     mode=0444
178 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/ManLint.pyc \
179     mode=0444
180 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Mapfile.py \
```



```
181     mode=0444
182 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/Mapfile.pyc \
183     mode=0444
184 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/ProcessCheck.py \
185     mode=0444
186 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/ProcessCheck.pyc \
187     mode=0444
188 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/SpellCheck.py \
189     mode=0444
190 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/SpellCheck.pyc \
191     mode=0444
192 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/__init__.py \
193     mode=0444
194 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Checks/__init__.pyc \
195     mode=0444
196 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Backup.py mode=0444
197 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Backup.pyc mode=0444
198 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Ignore.py mode=0444
199 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Ignore.pyc mode=0444
200 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Version.py mode=0444
201 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/Version.pyc \
202     mode=0444
203 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/WorkSpace.py \
204     mode=0444
205 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/WorkSpace.pyc \
206     mode=0444
207 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/__init__.py \
208     mode=0444
209 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/Scm/__init__.pyc \
210     mode=0444
211 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/__init__.py mode=0444
212 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/__init__.pyc mode=0444
213 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/hgext/__init__.py \
214     mode=0444
215 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/hgext/__init__.pyc \
216     mode=0444
217 file path=opt/onbld/lib/python$(PYTHON_VERSION)/onbld/hgext/cdm.py mode=0444
218 file path=opt/onbld/man/man1onbld/Install.1onbld
219 file path=opt/onbld/man/man1onbld/bldenv.1onbld
220 file path=opt/onbld/man/man1onbld/bringovercheck.1onbld
221 file path=opt/onbld/man/man1onbld/btxld.1onbld
222 file path=opt/onbld/man/man1onbld/cddlchk.1onbld
223 file path=opt/onbld/man/man1onbld/check_rtime.1onbld
224 file path=opt/onbld/man/man1onbld/checkpaths.1onbld
225 file path=opt/onbld/man/man1onbld/codereview.1onbld
226 file path=opt/onbld/man/man1onbld/cstyle.1onbld
227 file path=opt/onbld/man/man1onbld/cw.1onbld
228 file path=opt/onbld/man/man1onbld/find_elf.1onbld
229 file path=opt/onbld/man/man1onbld/findunref.1onbld
230 file path=opt/onbld/man/man1onbld/flg.flp.1onbld
241 file path=opt/onbld/man/man1onbld/get_depend_info.1onbld
231 file path=opt/onbld/man/man1onbld/git-pbchk.1onbld
232 file path=opt/onbld/man/man1onbld/hdrchk.1onbld
233 file path=opt/onbld/man/man1onbld/hgsetup.1onbld
234 file path=opt/onbld/man/man1onbld/interface_check.1onbld
235 file path=opt/onbld/man/man1onbld/interface_cmp.1onbld
236 file path=opt/onbld/man/man1onbld/jstyle.1onbld
237 file path=opt/onbld/man/man1onbld/lintdump.1onbld
249 file path=opt/onbld/man/man1onbld/make_pkg_db.1onbld
238 file path=opt/onbld/man/man1onbld/mapfilechk.1onbld
239 file path=opt/onbld/man/man1onbld/ndrgen.1onbld
240 file path=opt/onbld/man/man1onbld/nightly.1onbld
241 file path=opt/onbld/man/man1onbld/onu.1onbld
242 file path=opt/onbld/man/man1onbld/sccscheck.1onbld
243 file path=opt/onbld/man/man1onbld/signit.1onbld
244 file path=opt/onbld/man/man1onbld/signproto.1onbld
```

```
245 file path=opt/onbld/man/man1onbld/webrev.1onbld
246 file path=opt/onbld/man/man1onbld/which_scm.1onbld
247 file path=opt/onbld/man/man1onbld/ws.1onbld
248 file path=opt/onbld/man/man1onbld/wsdiff.1onbld
249 file path=opt/onbld/man/man1onbld/xref.1onbld
250 file path=opt/onbld/share/make.rules mode=0444
251 hardlink path=opt/onbld/bin/$(ARCH)/install.bin target=./install
252 legacy pkg=SUNWonbld desc="tools used to build the OS-Net consolidation" \
253     name="OS-Net Build Tools" version=11.11,REV=2009.10.22
254 license cr_Sun license=cr_Sun
255 license lic_CDDL license=lic_CDDL
256 license usr/src/tools/ctf/dwarf/THIRDPARTYLICENSE \
257     license=usr/src/tools/ctf/dwarf/THIRDPARTYLICENSE
258 license usr/src/tools/onbld/THIRDPARTYLICENSE \
259     license=usr/src/tools/onbld/THIRDPARTYLICENSE
260 link path=opt/onbld/bin/$(ARCH)/dmake target=make
261 link path=opt/onbld/bin/git-nits target=git-pbchk
262 link path=opt/onbld/lib/python target=python$(PYTHON_VERSION)
263 link path=opt/onbld/man/man1onbld/git-nits.1onbld target=git-pbchk.1onbld
264 # webrev(1) requires ps2pdf
265 depend fmri=print/filter/ghostscript type=require
266 # nightly(1) uses wget
267 depend fmri=web/wget type=require
```

```
**********************************************************
    3257 Thu Jan 25 19:54:35 2018
new/usr/src/tools/Makefile
8998 depcheck is useless and should be removed
8991 pmodes is useless and can be deleted
8990 /opt/onbld/gk is useless
**********************************************************
#
# 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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2016 Toomas Soome <tsoome@me.com>
# Copyright (c) 2016, Chris Fraire <cfraire@me.com>.
#

include ../Makefile.master

# Bootstrap problem --
# 'cw' must be built before anything else can be built.

BOOT_SUBDIRS= \
        cw

COMMON_SUBDIRS= \
        codereview \
        codesign \
        cscope-fast \
        ctf \
        depcheck \
        env \
        findunref \
        gk \
        install.bin \
        lintdump \
        make \
        ndrgen \
        onbld \
        pmodes \
        protocmp \
        protolist \
        scripts

#
#  special versions of commands for use only in build
#
UNSHIPPED_SUBDIRS = \
```



```
        localedef \
        mandoc \
        tic \
        vtfontcvt \
        zic

sparc_SUBDIRS= \
        chk4ubin \
        stabs \
        tokenize

i386_SUBDIRS=           \
        aw              \
        elfextract      \
        mbh_patch       \
        btxld

LINTSUBDIRS= \
        codereview \
        ctf \
        cw \
        findunref \
        lintdump \
        ndrgen \
        protocmp \
        protolist

SUBDIRS= \
        $($(MACH)_SUBDIRS) \
        $(COMMON_SUBDIRS) \
        $(UNSHIPPED_SUBDIRS)

include Makefile.tools

ROOTDIRS= \
        $(ROOTOPT) \
        $(ROOTONBLD) \
        $(ROOTONBLD)/bin \
        $(ROOTONBLD)/bin/$(MACH) \
        $(ROOTONBLD)/lib \
        $(ROOTONBLD)/lib/$(MACH) \
        $(ROOTONBLD)/lib/$(MACH)/64 \
        $(ROOTONBLD)/lib/perl \
        $(ROOTONBLD)/lib/python$(PYTHON_VERSION) \
        $(ROOTONBLD)/lib/python$(PYTHON_VERSION)/onbld \
        $(ROOTONBLD)/lib/python$(PYTHON_VERSION)/onbld/Checks \
        $(ROOTONBLD)/lib/python$(PYTHON_VERSION)/onbld/hgext \
        $(ROOTONBLD)/lib/python$(PYTHON_VERSION)/onbld/Scm \
        $(ROOTONBLD)/env \
        $(ROOTONBLD)/etc \
        $(ROOTONBLD)/etc/exception_lists \
        $(ROOTONBLD)/share \
        $(ROOTONBLD)/gk \
        $(ROOTONBLD)/man \
        $(ROOTONBLD)/man/man1onbld

all :=          TARGET= install
install :=      TARGET= install
clean :=        TARGET= clean
clobber :=      TARGET= clobber
lint :=         TARGET= lint
_msg :=         TARGET= _msg

.KEEP_STATE:

#
```

```
122 # Only create directories in the tools proto area when doing an actual
123 # build, not a clean or clobber.
124 #
125 DOROOTDIRS= $(ROOTDIRS)
126 clobber:= DOROOTDIRS=
127 clean:= DOROOTDIRS=

129 DOROOTONBLDLIBPY= $(ROOTONBLDLIBPY)
130 clobber:= DOROOTONBLDLIBPY=
131 clean:= DOROOTONBLDLIBPY=

133 all install: $(SUBDIRS)

135 clean: $(SUBDIRS)

137 clobber: $(SUBDIRS)
138         $(RM) -rf $(TOOLS_PROTO)

140 lint: $(LINTSUBDIRS)

142 _msg: $(MSGSUBDIRS)

144 .PARALLEL: $(SUBDIRS) $(CLOSED_SUBDIRS)

146 $(SUBDIRS) $(CLOSED_SUBDIRS): $(BOOT_SUBDIRS)

148 $(BOOT_SUBDIRS) $(SUBDIRS): $$(DOROOTDIRS) $$(DOROOTONBLDLIBPY) FRC
149         @cd $@; pwd; $(MAKE) $(TARGET)

151 $(ROOTDIRS):
152         $(INS.dir)

154 $(ROOTONBLDLIBPY): $(ROOTDIRS)
155         $(RM) -r $@; $(SYMLINK) python$(PYTHON_VERSION) $@

157 make: ctf

159 FRC:
```

```
**********************************************************
    2981 Thu Jan 25 19:54:36 2018
new/usr/src/tools/Makefile.tools
8990 /opt/onbld/gk is useless
**********************************************************
#
# 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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Definitions common to tool source.
#
include $(SRC)/Makefile.master

FILEMODE=       0555

TOOLS=                  $(SRC)/tools
TOOLS_PROTO=            $(TOOLS)/proto/root_$(MACH)-nd
ROOTOPT=                $(TOOLS_PROTO)/opt
ROOTONBLD=              $(ROOTOPT)/onbld
ROOTONBLDBIN=           $(ROOTONBLD)/bin
ROOTONBLDBINMACH=       $(ROOTONBLD)/bin/$(MACH)
ROOTONBLDETC=           $(ROOTONBLD)/etc
ROOTONBLDLIB=           $(ROOTONBLD)/lib
ROOTONBLDLIBMACH=       $(ROOTONBLD)/lib/$(MACH)
ROOTONBLDLIBMACH64=     $(ROOTONBLD)/lib/$(MACH)/64
ROOTONBLDLIBPERL=       $(ROOTONBLD)/lib/perl
ROOTONBLDLIBPY=         $(ROOTONBLD)/lib/python
ROOTONBLDENV=           $(ROOTONBLD)/env
ROOTONBLDGK=            $(ROOTONBLD)/gk
ROOTONBLDMAN=           $(ROOTONBLD)/man
ROOTONBLDMAN1ONBLD=     $(ROOTONBLD)/man/man1onbld
ROOTONBLDETCABI=        $(ROOTONBLD)/etc/abi
ROOTONBLDETCEXCEPT=     $(ROOTONBLD)/etc/exception_lists
ROOTONBLDSHARE=         $(ROOTONBLD)/share

CPPFLAGS=               -D_TS_ERRNO
ELFSIGN_O=              $(TRUE)
LDLIBS=
LDFLAGS=                $(MAPFILE.NES:%=-M%) $(MAPFILE.NED:%=-M%) \
                            $(MAPFILE.PGA:%=-M%)

ROOTONBLDPROG=          $(PROG:%=$(ROOTONBLDBIN)/%)
ROOTONBLDMACHPROG=      $(PROG:%=$(ROOTONBLDBINMACH)/%)
ROOTONBLDSHFILES=       $(SHFILES:%=$(ROOTONBLDBIN)/%)
ROOTONBLDMAKEFILES=     $(MAKEFILES:%=$(ROOTONBLDBIN)/%)
ROOTONBLDMACHSHFILES=   $(SHFILES:%=$(ROOTONBLDBINMACH)/%)
ROOTONBLDMACHBINARIES=  $(BINARIES:%=$(ROOTONBLDBINMACH)/%)
```



```
ROOTONBLDETCFILES=      $(ETCFILES:%=$(ROOTONBLDETC)/%)
ROOTONBLDENVFILES=      $(ENVFILES:%=$(ROOTONBLDENV)/%)
ROOTONBLDGKFILES=       $(GKFILES:%=$(ROOTONBLDGK)/.%)
ROOTONBLDGKSHFILES=     $(SHFILES:%=$(ROOTONBLDGK)/%)
ROOTONBLDPERLFILES=     $(PERLFILES:%=$(ROOTONBLDBIN)/%)
ROOTONBLDPERLMODULES=   $(PERLMODULES:%=$(ROOTONBLDLIBPERL)/%)
ROOTONBLDPYFILES=       $(PYFILES:%=$(ROOTONBLDBIN)/%)
ROOTONBLDMAN1ONBLDFILES=$(MAN1ONBLDFILES:%=$(ROOTONBLDMAN1ONBLD)/%)
ROOTONBLDABIAUDITFILES= $(ABI_AUDITFILES:%=$(ROOTONBLDETCABI)/%)
ROOTONBLDEXCEPTFILES=   $(EXCEPTFILES:%=$(ROOTONBLDETCEXCEPT)/%)

# Break a chicken-and-egg dependency cycle for the tools build
SCCSCHECK=@echo would sccscheck

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

$(ROOTONBLDETCEXCEPT)/%: $(CODEMGR_WS)/exception_lists/%
        $(INS.file)

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

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

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

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

$(ROOTONBLDMAN1ONBLD)/%: %
        $(INS.file)

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

$(ROOTONBLDGK)/.%: %
        $(INS.rename)

$(ROOTONBLDGK)/%: %
        $(INS.file)
```

```
**********************************************************
   10456 Thu Jan 25 19:54:37 2018
new/usr/src/tools/README.tools
8998 depcheck is useless and should be removed
8992 checkproto is useless and can be deleted
8991 pmodes is useless and can be deleted
8990 /opt/onbld/gk is useless
**********************************************************
```

```
#
# 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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.


This directory contains the tools used to do a full build of the
OS/Net workspace.  They usually live in the /opt/onbld directory on build
machines. From here, 'make install' will build and install the tools
in $ROOT/opt/onbld.
in $ROOT/opt/onbld. If you like, 'make pkg' will build the SUNWonbld
package in $(PKGARCHIVE). Installing that package will populate the
/opt/onbld directory, and create a root account for building called 'gk',
which uses csh and has a home directory of /opt/onbld/gk. You can
use this account to do full builds with 'nightly'. You don't have to,
but the 'gk' account has the path setup properly, has a .make.machines
file for dmake, and has a .login that sets up for dmake.

Layout of /opt/onbld
--------------------

/opt/onbld/etc/abi
        contains Solaris ABI database (ABI_*.db) and exceptions
        for ABI Auditing tool (interface_check, interface_cmp).

/opt/onbld/gk
        gk account's home directory.

/opt/onbld/bin
        basic bin directory - contains scripts.

/opt/onbld/bin/${MACH}
        architecture-specific bin directory for binaries.

/opt/onbld/env
        build environment files.

/opt/onbld/lib
        libraries used by the build tools.
```



```
/opt/onbld/lib/python<version>/
        python modules used by the build tools.

/opt/onbld/lib/python<version>/onbld/hgext
        Mercurial extensions.

/opt/onbld/lib/python/
        symlink to the modules directory of the currently preferred
        python version.  This exists to retain compatibility both for
        tools expecting only one supported version of python, and for
        user .hgrc files that expect to find cdm.py in
        /opt/onbld/lib/python/onbld/hgext.

/opt/onbld/man
        rudimentary man pages for some of the tools.


Tool Summary
------------

bldenv
        companion to 'nightly.' Takes the same environment file you
        used with 'nightly,' and starts a shell with the environment
        set up the same way as 'nightly' set it up. This is useful
        if you're trying to quickly rebuild portions of a workspace
        built by 'nightly'. 'ws' should not be used for this since it
        sets the environment up differently and may cause everything
        to rebuild (because of different -I or -L paths).

build_cscope
        builds cscope databases in the uts, the platform subdirectories
        of uts, and in usr/src. Uses cscope-fast.

cdm
        A Mercurial extension providing various commands useful for ON
        development

check_rtime
        checks ELF attributes used by ELF dynamic objects in the proto area.
        Used by 'nightly's -r option, to check a number of ELF runtime
        attributes for consistency with common build rules.  nightly uses
        the -o option to simplify the output for diffing with previous
        build results.  It also uses the -i option to obtain NEEDED and RUNPATH
        entries, which help detect changes in software dependencies and makes
        sure objects don't have any strange runpaths like /opt/SUNWspro/lib.

checkproto
        Runs protocmp and protolist on a workspace (or uses the environment
        variable CODEMGR_WS to determine the workspace). Checks the proto area
        against the packages.

codereview
        Given two filenames, creates a postscript file with the file
        differences highlighted.

codesign
        Tools for signing cryptographic modules using the official
        Sun release keys stored on a remote signing server. This
        directory contains signit, a client program for signing
        files with the signing server; signproto, a shell script
        that finds crypto modules in $ROOT and signs them using
        signit; and codesign_server.pl, the code that runs on the
        server. The codesign_server code is not used on an ON
        build machine but is kept here for source control purposes.

copyrightchk
```

```
110         Checks that files have appropriate SMI copyright notices.
111         Primarily used by wx

113 cscope-fast
114         The fast version of cscope that we use internally. Seems to work,
115         but may need more testing before it's placed in the gate. The source
116         just really needs to be here.
117
118 cstyle
119         checks C source for compliance with OS/Net guidelines.

121 ctfconvert
122         Convert symbolic debugging information in an object file to the Compact
123         ANSI-C Type Format (CTF).

125 ctfdump
126         Decode and display CTF data stored in a raw file or in an ELF file.

128 ctfmerge
129         Merge the CTF data from one or more object files.

145 depcheck
146         A tool to try an assess the dependencies of executables.  This tool
147         is not a definitive dependency check, but it does use "strings" and
148         "ldd" to gather as much information as it can.  The dependency check
149         tool can handle filenames and pkgnames.  Before using the dependency
150         checker you must build a database which reflects the properties and
151         files in your system.

131 elfcmp
132         Compares two ELF modules (e.g. .o files, executables) section by
133         section.  Useful for determining whether "trivial" changes -
134         cstyle, lint, etc - actually changed the code.  The -S option
135         is used to test whether two binaries are the same except for
136         the elfsign signature.

138 find_elf
139         Search a directory tree for ELF objects, and produce one line of
140         output per object. Used by check_rtime and interface_check to locate
141         the objects to examine.

143 findunref
144         Finds all files in a source tree that have access times older than a
145         certain time and are not in a specified list of exceptions.  Since
146         'nightly' timestamps the start of the build, and findunref uses its
147         timestamp (by default), this can be used to find all files that were
148         unreferenced during a nightly build).  Since some files are only used
149         during a SPARC or Intel build, 'findunref' needs to be run on
150         workspaces from both architectures and the results need to be merged.
151         For instance, if $INTELSRC and $SPARCSRC are set to the usr/src
152         directories of your Intel and SPARC nightly workspaces, then you
153         can merge the results like so:

155         $ findunref $INTELSRC $INTELSRC/tools/findunref/exception_list | \
156           sort > ~/unref-i386.out
157         $ findunref $SPARCSRC $SPARCSRC/tools/findunref/exception_list | \
158           sort > ~/unref-sparc.out
159         $ comm -12 ~/unref-i386.out ~/unref-sparc.out > ~/unref.out

161 hdrchk
162         checks headers for compliance with OS/Net standards (form, includes,
163         C++ guards).

165 hgsetup
166         creates a basic Mercurial configuration for the user.
```



```
168 hg-active
169         helper used by webrev to generate file lists for Mercurial
170         workspaces.

172 install.bin
173         binary version of /usr/sbin/install. Used to be vastly faster
174         (since /usr/sbin/install is a shell script), but may only be a bit
175         faster now. One speedup includes avoiding the name service for the
176         well-known, never-changing password entries like 'root' and 'sys.'

178 interface_check
179         detects and reports invalid versioning in ELF objects.
180         Optionally generates an interface description file for
181         the workspace.

183 interface_cmp
184         Compares two interface description files, as produced by
185         interface_check, and flags invalid deviations in ELF object
186         versioning between them. interface_cmp can be used between Solaris
187         gates to ensure that older releases remain compatible with the
188         development gate. It can also be used to validate new changes to
189         the development gate before they are integrated.

191 lintdump
192         dumps the contents of one or more lint libraries; see lintdump(1)

194 ndrgen
195         Network Data Language (NDL) RPC protocol compiler to support DCE
196         RPC/MSRPC and SMB/CIFS.  ndrgen takes an input protocol definition
197         file (say, proto.ndl) and generates an output C source file
198         (proto_ndr.c) containing the Network Data Representation (NDR)
199         marshalling routines to implement the RPC protocol.

201 nightly
202         nightly build script. Takes an environment (or 'env') file describing
203         such things as the workspace, the parent, and what to build. See
204         env/developer and env/gatekeeper for sample, hopefully well-commented
205         env files.

229 pmodes
230         enforces proper file ownership and permissions in pkgmap and package
231         prototype* files.  converts files if necessary

207 protocmp
208         compares proto lists and the package definitions. Used by nightly
209         to determine if the proto area matches the packages, and to detect
210         differences between a childs proto area and a parents.

212 protocmp.terse
213         transforms the output of protocmp into something a bit more friendly

215 protolist
216         create a list of what's in the proto area, to feed to protocmp.


219 ws
220         creates a shell with the environment set up to build in the given
221         workspace. Used mostly for non-full-build workspaces, so it sets up
222         to pull headers and libraries from the proto area of the parent if
223         they aren't in the childs proto area.

225 tokenize
226         Used to build the sun4u boot block.

228 webrev
229         Generates a set of HTML pages that show side-by-side diffs of
```

```
        changes in your workspace, for easy communication of code
        review materials.  Can automagically find edited files or use a
        manually-generated list; knows how to use wx's active file for
        lists of checked-out files and proposed SCCS comments.

which_scm
        Reports the current Source Code Management (SCM) system in use
        and the top-level directory of the workspace.

wsdiff
        Detect object differences between two ON proto areas. Used by
        nightly(1) to determine what changed between two builds. Handy
        for identifying the set of built objects impacted by a given
        source change. This information is needed for patch construction.


How to do a full build
----------------------

1. Find an environment file that might do what you want to do. If you're just
   a developer wanting to do a full build in a child of the gate, copy the
   'developer' environment file to a new name (private to you and/or the
   work being done in this workspace, to avoid collisions with others). Then
   edit the file and tailor it to your workspace. Remember that this file
   is a shell script, so it can do more than set environment variables.

2. Run 'nightly' and give it your environment file as an
2. Login as 'gk' (or root, but your PATH and .make.machines for dmake will
   not be right). Run 'nightly' and give it your environment file as an
   option. 'nightly' will first look for your environment file in
   /opt/onbld/env, and if it's not there then it will look for it as an
   absolute or relative path. Some people put their environment files in
   their workspace to keep them close.

3. When 'nightly' is complete, it will send a summary of what happened to
   $MAILTO. Usually, the less info in the mail the better. If you have failures,
   you can go look at the full log of what happened, generally in
   $CODEMGR_WS/log/log.<date>/nightly.log (the mail_msg it sent and the proto
   list are there too). You can also find the individual build logs, like
   'make clobber' and 'make install' output in $SRC, under names like
   clobber-${MACH}.out and install-${MACH}.out (for a DEBUG build). These
   will be smaller than nightly.log, and maybe more searchable.

Files you have to update to add a tool
--------------------------------------

1.  Add the tool in its appropriate place.
2.  Update the Makefile as required.
3.  Update usr/src/pkg/manifests/developer-build-onbld.mf
4.  Update usr/src/tools/README.tools (this file).
5.  Repeat 1-4 for any man pages.
```

```
**********************************************************
    3957 Thu Jan 25 19:54:44 2018
new/usr/src/tools/scripts/Makefile
8992 checkproto is useless and can be deleted
**********************************************************
```

```
#
# 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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright 2010, Richard Lowe

SHELL=/usr/bin/ksh93

SHFILES= \
        Install \
        bldenv \
        build_cscope \
        bringovercheck \
        checkpaths \
        checkproto \
        cstyle \
        elfcmp \
        flg.flp \
        genoffsets \
        hgsetup \
        nightly \
        onu \
        protocmp.terse \
        sccscheck \
        webrev \
        which_scm \
        ws \
        xref

PERLFILES= \
        check_rtime \
        find_elf \
        interface_check \
        interface_cmp \
        jstyle \
        validate_flg \
        validate_paths \
        wdiff

PERLMODULES= \
        onbld_elfmod.pm \
        onbld_elfmod_vertype.pm
```



```
PYFILES= \
        cddlchk \
        copyrightchk \
        git-pbchk \
        hdrchk \
        hg-active \
        mapfilechk \
        validate_pkg \
        wsdiff

SCRIPTLINKS= \
        git-nits

MAN1ONBLDFILES= \
        Install.1onbld \
        bldenv.1onbld \
        bringovercheck.1onbld \
        cddlchk.1onbld \
        checkpaths.1onbld \
        check_rtime.1onbld \
        cstyle.1onbld \
        find_elf.1onbld \
        flg.flp.1onbld \
        git-pbchk.1onbld \
        hdrchk.1onbld \
        interface_check.1onbld \
        interface_cmp.1onbld \
        hgsetup.1onbld \
        jstyle.1onbld \
        mapfilechk.1onbld \
        nightly.1onbld \
        onu.1onbld \
        sccscheck.1onbld \
        webrev.1onbld \
        which_scm.1onbld \
        ws.1onbld \
        wsdiff.1onbld \
        xref.1onbld

MAN1ONBLDLINKS= \
        git-nits.1onbld

MAKEFILES= \
        xref.mk

ETCFILES= \
        hgstyle \
        its.conf \
        its.reg

EXCEPTFILES= \
        check_rtime \
        interface_check \
        interface_cmp

CLEANFILES = $(SHFILES) $(PERLFILES) $(PYFILES) bldenv.1onbld onu.sh

onu.sh: onu.sh.in
        $(SED) -e "s:@PYTHON_VERSION@:$(PYTHON_VERSION):g" < onu.sh.in > $@

include ../Makefile.tools

ROOTONBLDSCRIPTLINKS = $(SCRIPTLINKS:%=$(ROOTONBLDBIN)/%)
ROOTONBLDMAN1ONBLDLINKS = $(MAN1ONBLDLINKS:%=$(ROOTONBLDMAN1ONBLD)/%)
```

```
128 $(ROOTONBLDETCFILES)            := FILEMODE=    644
129 $(ROOTONBLDEXCEPTFILES)         := FILEMODE=    644
130 $(ROOTONBLDPERLMODULES)         := FILEMODE=    644
131 $(ROOTONBLDMAKEFILES)           := FILEMODE=    644
132 $(ROOTONBLDMAN1ONBLDFILES)      := FILEMODE=    644

134 .KEEP_STATE:

136 all:    $(SHFILES) $(PERLFILES) $(PERLMODULES) $(PYFILES) \
137         $(MAN1ONBLDFILES) $(MAKEFILES)

139 $(ROOTONBLDBIN)/git-nits:
140         $(RM) $(ROOTONBLDBIN)/git-nits
141         $(SYMLINK) git-pbchk $(ROOTONBLDBIN)/git-nits

143 $(ROOTONBLDMAN1ONBLD)/git-nits.1onbld:
144         $(RM) $(ROOTONBLDMAN1ONBLD)/git-nits.1onbld
145         $(SYMLINK) git-pbchk.1onbld $(ROOTONBLDMAN1ONBLD)/git-nits.1onbld

147 install: all .WAIT $(ROOTONBLDSHFILES) $(ROOTONBLDPERLFILES)            \
148                 $(ROOTONBLDPERLMODULES) $(ROOTONBLDPYFILES)            \
149                 $(ROOTONBLDSCRIPTLINKS) $(ROOTONBLDMAN1ONBLDFILES)     \
150                 $(ROOTONBLDMAKEFILES) $(ROOTONBLDETCFILES)             \
151                 $(ROOTONBLDEXCEPTFILES) $(ROOTONBLDMAN1ONBLDLINKS)

153 clean:
154         $(RM) $(CLEANFILES)

156 bldenv: bldenv.sh stdenv.sh
157         $(RM) "$@"
158         sed -e '/# STDENV_START/ r stdenv.sh' bldenv.sh > "$@"
159         # Check for shell lint and fail if we hit warnings
160         shlintout="$$( /usr/bin/ksh93 -n "$@" 2>&1 )" ; \
161                 [[ "$${shlintout}" != "" ]] && \
162                 { print -r -- "$${shlintout}" ; false ; } || true
163         $(CHMOD) +x "$@"

165 bldenv.1onbld: bldenv
166         $(RM) "$@"
167         (set +o errexit ; ksh93 $? --nroff ; true) 2>&1 | \
168         sed -e 's/\.DS/.nf/g;s/\.DE/.fi/' \
169         -e 's/\.TH BLDENV 1/.TH BLDENV 1ONBLD/' \
170         -e 's/(1)/(1ONBLD)/' > "$@"

172 nightly: nightly.sh stdenv.sh
173         $(RM) "$@"
174         sed -e '/# STDENV_START/ r stdenv.sh' nightly.sh > nightly
175         $(CHMOD) +x "$@"

177 include ../Makefile.targ
```