```
**********************************************************
   35485 Tue Sep 11 12:46:14 2012
new/usr/src/Makefile.master
*** NO COMMENTS ***
**********************************************************
```

```
#
# 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) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012 by Delphix. All rights reserved.
#

#
# Makefile.master, global definitions for system source
#
ROOT=           /proto

#
# RELEASE_BUILD should be cleared for final release builds.
# NOT_RELEASE_BUILD is exactly what the name implies.
#
# INTERNAL_RELEASE_BUILD is a subset of RELEASE_BUILD. It mostly controls
# identification strings. Enabling RELEASE_BUILD automatically enables
# INTERNAL_RELEASE_BUILD.
#
# EXPORT_RELEASE_BUILD controls whether binaries are built in a form that
# can be released for export under a binary license.  It is orthogonal to
# the other *RELEASE_BUILD settings.  ("#" means do an export release
# build, "" means do a normal build.)
#
# CLOSED_BUILD controls whether we try to build files under
# usr/closed.  ("" means to build closed code, "#" means don't try to
# build it.)  Skipping the closed code implies doing an export release
# build.
#
# STRIP_COMMENTS toggles comment section striping. Generally the same setting
# as INTERNAL_RELEASE_BUILD.
#
# __GNUC toggles the building of ON components using gcc and related tools.
# Normally set to '#', set it to '' to do gcc build.
#
# The declaration POUND_SIGN is always '#'. This is needed to get around the
# make feature that '#' is always a comment delimiter, even when escaped or
# quoted. We use this macro expansion method to get POUND_SIGN rather than
# always breaking out a shell because the general case can cause a noticable
# slowdown in build times when so many Makefiles include Makefile.master.
#
```



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

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

NOT_RELEASE_BUILD=
INTERNAL_RELEASE_BUILD=                 $(POUND_SIGN)
RELEASE_BUILD=                          $(POUND_SIGN)
$(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
$(RELEASE_BUILD)INTERNAL_RELEASE_BUILD=
PATCH_BUILD=                            $(POUND_SIGN)

# If CLOSED_IS_PRESENT is not set, assume the closed tree is present.
CLOSED_BUILD_1= $(CLOSED_IS_PRESENT:yes=)
CLOSED_BUILD=   $(CLOSED_BUILD_1:no=$(POUND_SIGN))

EXPORT_RELEASE_BUILD=                   $(POUND_SIGN)
$(CLOSED_BUILD)EXPORT_RELEASE_BUILD=

# SPARC_BLD is '#' for an Intel build.
# INTEL_BLD is '#' for a Sparc build.
SPARC_BLD_1=    $(MACH:i386=$(POUND_SIGN))
SPARC_BLD=      $(SPARC_BLD_1:sparc=)
INTEL_BLD_1=    $(MACH:sparc=$(POUND_SIGN))
INTEL_BLD=      $(INTEL_BLD_1:i386=)

STRIP_COMMENTS= $(INTERNAL_RELEASE_BUILD)

# Are we building tonic closedbins? Unless you have used the
# -O flag to nightly or bldenv, leave the definition of TONICBUILD
# as $(POUND_SIGN).
#
# IF YOU CHANGE CLOSEDROOT, you MUST change install.bin
# to match the new definition.
TONICBUILD=     $(POUND_SIGN)
$(TONICBUILD)CLOSEDROOT= $(ROOT)-closed


# The variables below control the compilers used during the build.
# There are a number of permutations.
#
# __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
# one is not POUND_SIGN is the primary, with the other as the shadow.  They
# may also be used to control entirely compiler-specific Makefile assignments.
# __SUNC and Sun Studio are the default.
#
# __GNUC64 indicates that the 64bit build should use the GNU C compiler.
# There is no Sun C analogue.
#
# The following version-specific options are operative regardless of which
# compiler is primary, and control the versions of the given compilers to be
# used.  They also allow compiler-version specific Makefile fragments.
#
# __SSNEXT when set to the empty string enables options specific to the 'next'
# version of the Sun Studio compiler.
#
# __GNUC3 when the empty string uses and refers to GCC 3.x, it is the default.
# __GNUC4 when the empty string uses and refers to GCC 4.x.

__GNUC=                 $(POUND_SIGN)
$(__GNUC)__SUNC=        $(POUND_SIGN)
__GNUC64=               $(__GNUC)
```

```
__SSNEXT=               $(POUND_SIGN)

__GNUC3=
__GNUC4=                $(POUND_SIGN)
$(__GNUC4)__GNUC3=      $(POUND_SIGN)

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

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

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

JAVA_ROOT=      /usr/java

SFW_ROOT=       /usr/sfw
SFWINCDIR=      $(SFW_ROOT)/include
SFWLIBDIR=      $(SFW_ROOT)/lib
SFWLIBDIR64=    $(SFW_ROOT)/lib/$(MACH64)

$(__GNUC3)GCC_ROOT=     $(SFW_ROOT)
$(__GNUC4)GCC_ROOT=     /opt/gcc/4.4.4
GCCLIBDIR=      $(GCC_ROOT)/lib
GCCLIBDIR64=    $(GCC_ROOT)/lib/$(MACH64)

RPCGEN=         /usr/bin/rpcgen
STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
ECHO=           echo
INS=            install
TRUE=           true
SYMLINK=        /usr/bin/ln -s
LN=             /usr/bin/ln
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
NAWK=           /usr/bin/nawk
CP=             /usr/bin/cp -f
MCS=            /usr/ccs/bin/mcs
CAT=            /usr/bin/cat
ELFDUMP=        /usr/ccs/bin/elfdump
M4=             /usr/ccs/bin/m4
STRIP=          /usr/ccs/bin/strip
LEX=            /usr/ccs/bin/lex
FLEX=           $(SFW_ROOT)/bin/flex
YACC=           /usr/ccs/bin/yacc
CPP=            /usr/lib/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
NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
CTFCVTPTBL=     $(ONBLD_TOOLS)/bin/ctfcvtptbl
CTFFINDMOD=     $(ONBLD_TOOLS)/bin/ctffindmod
XREF=           $(ONBLD_TOOLS)/bin/xref
FIND=           /usr/bin/find
PERL=           /usr/bin/perl
PYTHON_24=      /usr/bin/python2.4
PYTHON_26=      /usr/bin/python2.6
PYTHON=         $(PYTHON_26)
PYTHON=         $(PYTHON_24)
SORT=           /usr/bin/sort
TOUCH=          /usr/bin/touch
WC=             /usr/bin/wc
XARGS=          /usr/bin/xargs
ELFEDIT=        /usr/bin/elfedit
ELFSIGN=        /usr/bin/elfsign
DTRACE=         /usr/sbin/dtrace -xnolibs
UNIQ=           /usr/bin/uniq
TAR=            /usr/bin/tar

FILEMODE=       644
DIRMODE=        755

#
# The version of the patch makeup table optimized for build-time use.  Used
# during patch builds only.
$(PATCH_BUILD)PMTMO_FILE=$(SRC)/patch_makeup_table.mo

# 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 $<

#
# IMPORTANT:: If you change any of INS.file, INS.dir, INS.rename,
```

```
258 # INS.link or INS.symlink here, then you must also change the
259 # corresponding override definitions in $CLOSED/Makefile.tonic.
260 # If you do not do this, then the closedbins build for the OpenSolaris
261 # community will break. PS, the gatekeepers will be upset too.
262 INS.file=       $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
263 INS.dir=        $(INS) -s -d -m $(DIRMODE) $@
264 # installs and renames at once
265 #
266 INS.rename=     $(INS.file); $(MV) $(@D)/$(<F) $@

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

273 #
274 # Python bakes the mtime of the .py file into the compiled .pyc and
275 # rebuilds if the baked-in mtime != the mtime of the source file
276 # (rather than only if it's less than), thus when installing python
277 # files we must make certain to not adjust the mtime of the source
278 # (.py) file.
279 #
280 INS.pyfile=     $(INS.file); $(TOUCH) -r $< $@

282 # MACH must be set in the shell environment per uname -p on the build host
283 # More specific architecture variables should be set in lower makefiles.
284 #
285 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
286 # architectures on which we do not build 64-bit versions.
287 # (There are no such architectures at the moment.)
288 #
289 # Set BUILD64=# in the environment to disable 64-bit amd64
290 # builds on i386 machines.

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

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

298 sparc_BUILD64=
299 i386_BUILD64=
300 BUILD64=        $($(MACH)_BUILD64)

302 #
303 # C compiler mode. Future compilers may change the default on us,
304 # so force extended ANSI mode globally. Lower level makefiles can
305 # override this by setting CCMODE.
306 #
307 CCMODE=                 -Xa
308 CCMODE64=               -Xa

310 #
311 # C compiler verbose mode. This is so we can enable it globally,
312 # but turn it off in the lower level makefiles of things we cannot
313 # (or aren't going to) fix.
314 #
315 CCVERBOSE=              -v

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

321 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
322 # symbols (used to detect conflicts between objects that use global registers)
323 # we disable this now for safety, and because genunix doesn't link with
```



```
324 # this feature (the v9 default) enabled.
325 #
326 # REGSYM is separate since the C++ driver syntax is different.
327 CCREGSYM=               -Wc,-Qiselect-regsym=0
328 CCCREGSYM=              -Qoption cg -Qiselect-regsym=0

330 # Prevent the removal of static symbols by the SPARC code generator (cg).
331 # The x86 code generator (ube) does not remove such symbols and as such
332 # using this workaround is not applicable for x86.
333 #
334 CCSTATICSYM=            -Wc,-Qassembler-ounrefsym=0
335 #
336 # generate 32-bit addresses in the v9 kernel. Saves memory.
337 CCABS32=                -Wc,-xcode=abs32
338 #
339 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
340 # system calls.
341 $(__GNUC4)CC32BITCALLERS=               -_gcc=-massume-32bit-callers

343 # GCC, especially, is increasingly beginning to auto-inline functions and
344 # sadly does so separately not under the general -fno-inline-functions
345 $(__GNUC4)CCNOAUTOINLINE= -_gcc=-fno-inline-small-functions \
346         -_gcc=-fno-inline-functions-called-once

348 # One optimization the compiler might perform is to turn this:
349 #       #pragma weak foo
350 #       extern int foo;
351 #       if (&foo)
352 #               foo = 5;
353 # into
354 #       foo = 5;
355 # Since we do some of this (foo might be referenced in common kernel code
356 # but provided only for some cpu modules or platforms), we disable this
357 # optimization.
358 #
359 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
360 i386_CCUNBOUND  =
361 CCUNBOUND       = $($(MACH)_CCUNBOUND)

363 #
364 # compiler '-xarch' flag. This is here to centralize it and make it
365 # overridable for testing.
366 sparc_XARCH=    -m32
367 sparcv9_XARCH=  -m64
368 i386_XARCH=
369 amd64_XARCH=    -m64 -Ui386 -U__i386

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

377 #
378 # These flags define what we need to be 'standalone' i.e. -not- part
379 # of the rather more cosy userland environment.  This basically means
380 # the kernel.
381 #
382 # XX64  future versions of gcc will make -mcmodel=kernel imply -mno-red-zone
383 #
384 sparc_STAND_FLAGS=      -_gcc=-ffreestanding
385 sparcv9_STAND_FLAGS=    -_gcc=-ffreestanding
386 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
387 # additions to SSE (SSE2, AVX ,etc.)
388 NO_SIMD=                -_gcc=-mno-mmx -_gcc=-mno-sse
389 i386_STAND_FLAGS=       -_gcc=-ffreestanding $(NO_SIMD)
```

```
390 amd64_STAND_FLAGS=      -xmodel=kernel $(NO_SIMD)

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

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

398 #
399 # disable the incremental linker
400 ILDOFF=                  -xildoff
401 #
402 XDEPEND=                 -xdepend
403 XFFLAG=                  -xF=%all
404 XESS=                    -xs
405 XSTRCONST=               -xstrconst

407 #
408 # turn warnings into errors (C)
409 CERRWARN = -errtags=yes -errwarn=%all
410 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
411 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

413 # Normally cw(1) would translate -v into a set of options including these
414 # but as they're GCC 4.x specific, we can't do that
415 $(__GNUC4)CERRWARN +=   -_gcc=-Wno-address -_gcc=-Wno-array-bounds

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

421 # C99 mode
422 C99_ENABLE=     -xc99=%all
423 C99_DISABLE=    -xc99=%none
424 C99MODE=        $(C99_DISABLE)
425 C99LMODE=       $(C99MODE:-xc99%=-Xc99%)

427 # In most places, assignments to these macros should be appended with +=
428 # (CPPFLAGS.master 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 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
476 CTF_FLAGS_i386  = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
477 CTF_FLAGS       = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)

479 #
480 # Flags used with genoffsets
481 #
482 GOFLAGS = -_noecho \
483         $(CALLSYMS) \
484         $(CDWARFSTR)

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

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

492 #
493 # tradeoff time for space (smaller is better)
494 #
495 sparc_SPACEFLAG         = -xspace -W0,-Lt
496 sparcv9_SPACEFLAG       = -xspace -W0,-Lt
497 i386_SPACEFLAG          = -xspace
498 amd64_SPACEFLAG         =

500 SPACEFLAG               = $($(MACH)_SPACEFLAG)
501 SPACEFLAG64             = $($(MACH64)_SPACEFLAG)

503 #
504 # The Sun Studio 11 compiler has changed the behaviour of integer
505 # wrap arounds and so a flag is needed to use the legacy behaviour
506 # (without this flag panics/hangs could be exposed within the source).
507 #
508 sparc_IROPTFLAG         = -W2,-xwrap_int
509 sparcv9_IROPTFLAG       = -W2,-xwrap_int
510 i386_IROPTFLAG          =
511 amd64_IROPTFLAG         =

513 IROPTFLAG               = $($(MACH)_IROPTFLAG)
514 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)

516 sparc_XREGSFLAG         = -xregs=no%appl
517 sparcv9_XREGSFLAG       = -xregs=no%appl
518 i386_XREGSFLAG          =
519 amd64_XREGSFLAG         =

521 XREGSFLAG               = $($(MACH)_XREGSFLAG)
```

```
522 XREGSFLAG64             = $($(MACH64)_XREGSFLAG)

524 CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
525                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
526                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE)
527 CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
528                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
529                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE)
530 #
531 # Flags that are used to build parts of the code that are subsequently
532 # run on the build machine (also known as the NATIVE_BUILD).
533 #
534 NATIVE_CFLAGS=  $(COPTFLAG) $($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
535                 $(ILDOFF) $(CERRWARN) $(C99MODE) $($(NATIVE_MACH)_CCUNBOUND) \
536                 $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE)

538 DTEXTDOM=-DTEXT_DOMAIN=\"$(TEXT_DOMAIN)\"       # For messaging.
539 DTS_ERRNO=-D_TS_ERRNO
540 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
541         $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
542 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
543 CPPFLAGS=       $(CPPFLAGS.master)
544 AS_CPPFLAGS=    $(CPPFLAGS.master)
545 JAVAFLAGS=      -deprecation

547 #
548 # For source message catalogue
549 #
550 .SUFFIXES: $(SUFFIXES) .i .po
551 MSGROOT= $(ROOT)/catalog
552 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
553 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
554 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
555 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

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

569 #
570 # This is overwritten by local Makefile when PROG is a list.
571 #
572 POFILE= $(PROG).po

574 sparc_CCFLAGS=          -cg92 -compat=4 \
575                         -Qoption ccfe -messages=no%anachronism \
576                         $(CCERRWARN)
577 sparcv9_CCFLAGS=        $(sparcv9_XARCH) -dalign -compat=5 \
578                         -Qoption ccfe -messages=no%anachronism \
579                         -Qoption ccfe -features=no%conststrings \
580                         $(CCCREGSYM) \
581                         $(CCERRWARN)
582 i386_CCFLAGS=           -compat=4 \
583                         -Qoption ccfe -messages=no%anachronism \
584                         -Qoption ccfe -features=no%conststrings \
585                         $(CCERRWARN)
586 amd64_CCFLAGS=          $(amd64_XARCH) -compat=5 \
587                         -Qoption ccfe -messages=no%anachronism \
```



```
588                         -Qoption ccfe -features=no%conststrings \
589                         $(CCERRWARN)

591 sparc_CCOPTFLAG=        -O
592 sparcv9_CCOPTFLAG=      -O
593 i386_CCOPTFLAG=         -O
594 amd64_CCOPTFLAG=        -O

596 CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
597 CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
598 CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS)
599 CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS)

601 #
602 #
603 #
604 ELFWRAP_FLAGS   =
605 ELFWRAP_FLAGS64 =       -64

607 #
608 # Various mapfiles that are used throughout the build, and delivered to
609 # /usr/lib/ld.
610 #
611 MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
612 MAPFILE.NED_sparc =
613 MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
614 MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
615 MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
616 MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
617 MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy

619 #
620 # Generated mapfiles that are compiler specific, and used throughout the
621 # build.  These mapfiles are not delivered in /usr/lib/ld.
622 #
623 MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
624 $(__GNUC64)MAPFILE.NGB_sparc= \
625                         $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
626 MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
627 $(__GNUC64)MAPFILE.NGB_sparcv9= \
628                         $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
629 MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
630 $(__GNUC64)MAPFILE.NGB_i386= \
631                         $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
632 MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
633 $(__GNUC64)MAPFILE.NGB_amd64= \
634                         $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
635 MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

637 #
638 # A generic interface mapfile name, used by various dynamic objects to define
639 # the interfaces and interposers the object must export.
640 #
641 MAPFILE.INT =           mapfile-intf

643 #
644 # LDLIBS32 can be set in the environment to override the following assignment.
645 # LDLIBS64 can be set to override the assignment made in Makefile.master.64.
646 # These environment settings make sure that no libraries are searched outside
647 # of the local workspace proto area:
648 #       LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
649 #       LDLIBS64=-YP,$ROOT/lib/$MACH64:$ROOT/usr/lib/$MACH64
650 #
651 LDLIBS32 =      $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
652 LDLIBS.cmd =    $(LDLIBS32)
653 LDLIBS.lib =    $(LDLIBS32)
```

```
654 #
655 # Define compilation macros.
656 #
657 COMPILE.c=      $(CC) $(CFLAGS) $(CPPFLAGS) -c
658 COMPILE64.c=    $(CC) $(CFLAGS64) $(CPPFLAGS) -c
659 COMPILE.cc=     $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
660 COMPILE64.cc=   $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
661 COMPILE.s=      $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
662 COMPILE64.s=    $(AS) $(ASFLAGS) $($(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
663 COMPILE.d=      $(DTRACE) -G -32
664 COMPILE64.d=    $(DTRACE) -G -64
665 COMPILE.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
666 COMPILE64.b=    $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

668 CLASSPATH=      .
669 COMPILE.java=   $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

671 #
672 # Link time macros
673 #
674 CCNEEDED                = -lC
675 CCEXTNEEDED             = -lCrun -lCstd
676 $(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -R$(GCCLIBDIR) -lstdc++ -lgcc_s
677 $(__GNUC)CCEXTNEEDED    = $(CCNEEDED)

679 LINK.c=         $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
680 LINK64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
681 NORUNPATH=      -norunpath -nolib
682 LINK.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
683                 $(LDFLAGS) $(CCNEEDED)
684 LINK64.cc=      $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
685                 $(LDFLAGS) $(CCNEEDED)

687 #
688 # lint macros
689 #
690 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
691 # ON is built with a version of lint that has the fix for 4484186.
692 #
693 ALWAYS_LINT_DEFS =      -errtags=yes -s
694 ALWAYS_LINT_DEFS +=     -erroff=E_PTRDIFF_OVERFLOW
695 ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_NARROW_CONV
696 ALWAYS_LINT_DEFS +=     -U__PRAGMA_REDEFINE_EXTNAME
697 ALWAYS_LINT_DEFS +=     $(C99LMODE)
698 ALWAYS_LINT_DEFS +=     -errsecurity=$(SECLEVEL)
699 ALWAYS_LINT_DEFS +=     -erroff=E_SEC_CREAT_WITHOUT_EXCL
700 ALWAYS_LINT_DEFS +=     -erroff=E_SEC_FORBIDDEN_WARN_CREAT
701 # XX64 -- really only needed for amd64 lint
702 ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_INT_TO_SMALL_INT
703 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_CONST_TO_SMALL_INT
704 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_TO_SMALL_INT
705 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_TO_PTR_FROM_INT
706 ALWAYS_LINT_DEFS +=     -erroff=E_COMP_INT_WITH_LARGE_INT
707 ALWAYS_LINT_DEFS +=     -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
708 ALWAYS_LINT_DEFS +=     -erroff=E_PASS_INT_TO_SMALL_INT
709 ALWAYS_LINT_DEFS +=     -erroff=E_PTR_CONV_LOSES_BITS

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

715 SECLEVEL=       core
716 LINT.c=         $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
717                 $(ALWAYS_LINT_DEFS)
718 LINT64.c=       $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
719                 $(ALWAYS_LINT_DEFS)
```



```
720 LINT.s=         $(LINT.c)

722 # For some future builds, NATIVE_MACH and MACH might be different.
723 # Therefore, NATIVE_MACH needs to be redefined in the
724 # environment as ‘uname -p‘ to override this macro.
725 #
726 # For now at least, we cross-compile amd64 on i386 machines.
727 NATIVE_MACH=    $(MACH:amd64=i386)

729 # Define native compilation macros
730 #

732 # Base directory where compilers are loaded.
733 # Defined here so it can be overridden by developer.
734 #
735 SPRO_ROOT=              $(BUILD_TOOLS)/SUNWspro
736 SPRO_VROOT=             $(SPRO_ROOT)/SS12
737 GNU_ROOT=               $(SFW_ROOT)

739 # Till SS12u1 formally becomes the NV CBE, LINT is hard
740 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
741 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
742 # i386_LINT, amd64_LINT.
743 # Reset them when SS12u1 is rolled out.
744 #

746 # Specify platform compiler versions for languages
747 # that we use (currently only c and c++).
748 #
749 sparc_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
750 $(__GNUC)sparc_CC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
751 sparc_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
752 $(__GNUC)sparc_CCC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
753 sparc_CPP=              /usr/ccs/lib/cpp
754 sparc_AS=               /usr/ccs/bin/as -xregsym=no
755 sparc_LD=               /usr/ccs/bin/ld
756 sparc_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

758 sparcv9_CC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
759 $(__GNUC64)sparcv9_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
760 sparcv9_CCC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
761 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
762 sparcv9_CPP=            /usr/ccs/lib/cpp
763 sparcv9_AS=             /usr/ccs/bin/as -xregsym=no
764 sparcv9_LD=             /usr/ccs/bin/ld
765 sparcv9_LINT=           $(SPRO_ROOT)/sunstudio12.1/bin/lint

767 i386_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
768 $(__GNUC)i386_CC=       $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
769 i386_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
770 $(__GNUC)i386_CCC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
771 i386_CPP=              /usr/ccs/lib/cpp
772 i386_AS=               /usr/ccs/bin/as
773 $(__GNUC)i386_AS=       $(ONBLD_TOOLS)/bin/$(MACH)/aw
774 i386_LD=               /usr/ccs/bin/ld
775 i386_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

777 amd64_CC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
778 $(__GNUC64)amd64_CC=    $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
779 amd64_CCC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
780 $(__GNUC64)amd64_CCC=   $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
781 amd64_CPP=             /usr/ccs/lib/cpp
782 amd64_AS=              $(ONBLD_TOOLS)/bin/$(MACH)/aw
783 amd64_LD=              /usr/ccs/bin/ld
784 amd64_LINT=            $(SPRO_ROOT)/sunstudio12.1/bin/lint
```

```
786 NATIVECC=               $($(NATIVE_MACH)_CC)
787 NATIVECCC=              $($(NATIVE_MACH)_CCC)
788 NATIVECPP=              $($(NATIVE_MACH)_CPP)
789 NATIVEAS=               $($(NATIVE_MACH)_AS)
790 NATIVELD=               $($(NATIVE_MACH)_LD)
791 NATIVELINT=             $($(NATIVE_MACH)_LINT)

793 #
794 # Makefile.master.64 overrides these settings
795 #
796 CC=                     $(NATIVECC)
797 CCC=                    $(NATIVECCC)
798 CPP=                    $(NATIVECPP)
799 AS=                     $(NATIVEAS)
800 LD=                     $(NATIVELD)
801 LINT=                   $(NATIVELINT)

803 # The real compilers used for this build
804 CW_CC_CMD=              $(CC) -_compiler
805 CW_CCC_CMD=             $(CCC) -_compiler
806 REAL_CC=                $(CW_CC_CMD:sh)
807 REAL_CCC=               $(CW_CCC_CMD:sh)

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

812 BDIRECT=        -Bdirect
813 BDYNAMIC=       -Bdynamic
814 BLOCAL=         -Blocal
815 BNODIRECT=      -Bnodirect
816 BREDUCE=        -Breduce
817 BSTATIC=        -Bstatic

819 ZDEFS=          -zdefs
820 ZDIRECT=        -zdirect
821 ZIGNORE=        -zignore
822 ZINITFIRST=     -zinitfirst
823 ZINTERPOSE=     -zinterpose
824 ZLAZYLOAD=      -zlazyload
825 ZLOADFLTR=      -zloadfltr
826 ZMULDEFS=       -zmuldefs
827 ZNODEFAULTLIB=  -znodefaultlib
828 ZNODEFS=        -znodefs
829 ZNODELETE=      -znodelete
830 ZNODLOPEN=      -znodlopen
831 ZNODUMP=        -znodump
832 ZNOLAZYLOAD=    -znolazyload
833 ZNOLDYNSYM=     -znoldynsym
834 ZNORELOC=       -znoreloc
835 ZNOVERSION=     -znoversion
836 ZRECORD=        -zrecord
837 ZREDLOCSYM=     -zredlocsym
838 ZTEXT=          -ztext
839 ZVERBOSE=       -zverbose

841 GSHARED=        -G
842 CCMT=           -mt

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

847 sparc_C_PICFLAGS =      -K pic
848 sparcv9_C_PICFLAGS =    -K pic
849 i386_C_PICFLAGS =       -K pic
850 amd64_C_PICFLAGS =      -K pic
851 C_PICFLAGS =            $($(MACH)_C_PICFLAGS)
```



```
852 C_PICFLAGS64 =          $($(MACH64)_C_PICFLAGS)

854 sparc_C_BIGPICFLAGS =   -K PIC
855 sparcv9_C_BIGPICFLAGS = -K PIC
856 i386_C_BIGPICFLAGS =    -K PIC
857 amd64_C_BIGPICFLAGS =   -K PIC
858 C_BIGPICFLAGS =         $($(MACH)_C_BIGPICFLAGS)
859 C_BIGPICFLAGS64 =       $($(MACH64)_C_BIGPICFLAGS)

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

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

872 #
873 # Default label for CTF sections
874 #
875 CTFCVTFLAGS=            -i -L VERSION

877 #
878 # Override to pass module-specific flags to ctfmerge.  Currently used
879 # only by krtld to turn on fuzzy matching.
880 #
881 CTFMRGFLAGS=

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

885 ELFSIGN_O=      $(TRUE)
886 ELFSIGN_CRYPTO= $(ELFSIGN_O)
887 ELFSIGN_OBJECT= $(ELFSIGN_O)
888 $(EXPORT_RELEASE_BUILD)ELFSIGN_O =      $(ELFSIGN)
889 $(EXPORT_RELEASE_BUILD)ELFSIGN_CFNAME = SUNWosnetCF
890 $(EXPORT_RELEASE_BUILD)ELFSIGN_KEY =    \
891                         $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_CFNAME)
892 $(EXPORT_RELEASE_BUILD)ELFSIGN_CERT=    \
893                         $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_CFNAME)
894 $(EXPORT_RELEASE_BUILD)ELFSIGN_SENAME = SUNWosnetSE
895 $(EXPORT_RELEASE_BUILD)ELFSIGN_SEKEY =  \
896                         $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_SENAME)
897 $(EXPORT_RELEASE_BUILD)ELFSIGN_SECERT=  \
898                         $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_SENAME)
899 $(EXPORT_RELEASE_BUILD)ELFSIGN_CRYPTO=  $(ELFSIGN_O) sign \
900                         $(ELFSIGN_FORMAT_OPTION) \
901                         -k $(ELFSIGN_KEY) -c $(ELFSIGN_CERT) -e $@
902 $(EXPORT_RELEASE_BUILD)ELFSIGN_OBJECT=  $(ELFSIGN_O) sign \
903                         $(ELFSIGN_FORMAT_OPTION) \
904                         -k $(ELFSIGN_SEKEY) -c $(ELFSIGN_SECERT) -e $@

906 # Rules (normally from make.rules) and macros which are used for post
907 # processing files. Normally, these do stripping of the comment section
908 # automatically.
909 #    RELEASE_CM:        Should be editted to reflect the release.
910 #    POST_PROCESS_O:    Post-processing for '.o' files.
911 #    POST_PROCESS_A:    Post-processing for '.a' files (currently null).
912 #    POST_PROCESS_SO:   Post-processing for '.so' files.
913 #    POST_PROCESS:      Post-processing for executable files (no suffix).
914 # Note that these macros are not completely generalized as they are to be
915 # used with the file name to be processed following.
916 #
917 # 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} -c -a $(RELEASE_CM) -a $(DEV_CM)
$(STRIP_COMMENTS)PROCESS_COMMENT=   @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
$(RELEASE_BUILD)PROCESS_COMMENT=    @?${MCS} -d -a $(RELEASE_CM)

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

POST_PROCESS_O=         $(PROCESS_COMMENT) $@
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
```

```
1051 .bin.o:
1052         $(COMPILE.b) -o $@ $<
1053         $(POST_PROCESS_O)

1055 .java.class:
1056         $(COMPILE.java) $<

1058 # Bourne and Korn shell script message catalog build rules.
1059 # We extract all gettext strings with sed(1) (being careful to permit
1060 # multiple gettext strings on the same line), weed out the dups, and
1061 # build the catalogue with awk(1).

1063 .sh.po .ksh.po:
1064         $(SED) -n -e ":a"                                       \
1065                   -e "h"                                        \
1066                   -e "s/.*gettext *\(\"[^\"]*\"\).*/\1/p"       \
1067                   -e "x"                                        \
1068                   -e "s/\(.*\)gettext *\"[^\"]*\"\(.*\)/\1\2/"  \
1069                   -e "t a"                                      \
1070                 $< | sort -u | awk '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1072 #
1073 # Python and Perl executable and message catalog build rules.
1074 #
1075 .SUFFIXES: .pl .pm .py .pyc

1077 .pl:
1078         $(RM) $@;
1079         $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1080         $(CHMOD) +x $@

1082 .py:
1083         $(RM) $@; $(CAT) $< > $@; $(CHMOD) +x $@

1085 .py.pyc:
1086         $(RM) $@
1087         $(PYTHON) -mpy_compile $<
1088         @[ $(<)c = $@ ] || $(MV) $(<)c $@

1090 .py.po:
1091         $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;

1093 .pl.po .pm.po:
1094         $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1095         $(RM)   $@ ;
1096         $(SED) "/^domain/d" < $(<F).po > $@ ;
1097         $(RM) $(<F).po

1099 #
1100 # When using xgettext, we want messages to go to the default domain,
1101 # rather than the specified one.  This special version of the
1102 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1103 # causing xgettext to put all messages into the default domain.
1104 #
1105 CPPFORPO=$(COMPILE.cpp:\"$(TEXT_DOMAIN)\"=TEXT_DOMAIN)

1107 .c.i:
1108         $(CPPFORPO) $< > $@

1110 .h.i:
1111         $(CPPFORPO) $< > $@

1113 .y.i:
1114         $(YACC) -d $<
1115         $(CPPFORPO) y.tab.c  > $@
```



```
1116         $(RM) y.tab.c

1118 .l.i:
1119         $(LEX) $<
1120         $(CPPFORPO) lex.yy.c  > $@
1121         $(RM) lex.yy.c

1123 .c.po:
1124         $(CPPFORPO) $< > $<.i
1125         $(BUILD.po)

1127 .y.po:
1128         $(YACC) -d $<
1129         $(CPPFORPO) y.tab.c  > $<.i
1130         $(BUILD.po)
1131         $(RM) y.tab.c

1133 .l.po:
1134         $(LEX) $<
1135         $(CPPFORPO) lex.yy.c  > $<.i
1136         $(BUILD.po)
1137         $(RM) lex.yy.c

1139 #
1140 # Rules to perform stylistic checks
1141 #
1142 .SUFFIXES: .x .xml .check .xmlchk

1144 .h.check:
1145         $(DOT_H_CHECK)

1147 .x.check:
1148         $(DOT_X_CHECK)

1150 .xml.xmlchk:
1151         $(MANIFEST_CHECK)

1153 #
1154 # Rules to process ONC+ Source partial files
1155 #
1156 %_onc_plus:     %
1157         @$(ECHO) "extracting code from $< ... "
1158         sed -n -e '/ONC_PLUS EXTRACT START/,/ONC_PLUS EXTRACT END/p' $<  > $@

1160 #
1161 # Include rules to render automated sccs get rules "safe".
1162 #
1163 include $(SRC)/Makefile.noget
```

```
**********************************************************
   10813 Tue Sep 11 12:46:15 2012
new/usr/src/pkg/manifests/developer-build-onbld.mf
*** NO COMMENTS ***
**********************************************************
#
# 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 2010, Richard Lowe
# Copyright 2012, Piotr Jasiukajtis
#

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/perl
dir path=opt/onbld/lib/python2.4
dir path=opt/onbld/lib/python2.4/onbld
dir path=opt/onbld/lib/python2.4/onbld/Checks
dir path=opt/onbld/lib/python2.4/onbld/Scm
dir path=opt/onbld/lib/python2.4/onbld/hgext
dir path=opt/onbld/lib/python2.6
dir path=opt/onbld/lib/python2.6/onbld
dir path=opt/onbld/lib/python2.6/onbld/Checks
dir path=opt/onbld/lib/python2.6/onbld/Scm
dir path=opt/onbld/lib/python2.6/onbld/hgext
dir path=opt/onbld/man
```



```
dir path=opt/onbld/man/man1
$(i386_ONLY)file path=opt/onbld/bin/$(ARCH)/aw 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)/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
$(i386_ONLY)file path=opt/onbld/bin/$(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
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/tokenize mode=0555
$(sparc_ONLY)file path=opt/onbld/bin/$(ARCH)/tokenize.exe mode=0555
file path=opt/onbld/bin/Install mode=0555
file path=opt/onbld/bin/bindrop 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/cryptodrop mode=0555
file path=opt/onbld/bin/cstyle mode=0555
file path=opt/onbld/bin/ctfcvtptbl mode=0555
file path=opt/onbld/bin/ctffindmod mode=0555
file path=opt/onbld/bin/elfcmp mode=0555
file path=opt/onbld/bin/elfsigncmp 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/mkreadme_osol mode=0555
file path=opt/onbld/bin/mktpl 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
file path=opt/onbld/bin/validate_paths mode=0555
file path=opt/onbld/bin/validate_pkg mode=0555
```

```
123 file path=opt/onbld/bin/wdiff mode=0555
124 file path=opt/onbld/bin/webrev mode=0555
125 file path=opt/onbld/bin/which_scm mode=0555
126 file path=opt/onbld/bin/ws mode=0555
127 file path=opt/onbld/bin/wsdiff mode=0555
128 file path=opt/onbld/bin/xref mode=0555
129 file path=opt/onbld/bin/xref.mk
130 file path=opt/onbld/env/developer
131 file path=opt/onbld/env/gatekeeper
132 file path=opt/onbld/env/illumos
133 file path=opt/onbld/etc/SampleLinks
134 file path=opt/onbld/etc/SamplePkgLinks
135 file path=opt/onbld/etc/exception_lists/check_rtime
136 file path=opt/onbld/etc/exception_lists/interface_check
137 file path=opt/onbld/etc/exception_lists/interface_cmp
138 file path=opt/onbld/etc/hgstyle
139 file path=opt/onbld/etc/its.conf
140 file path=opt/onbld/etc/its.reg
141 file path=opt/onbld/gk/.cshrc
142 file path=opt/onbld/gk/.login
143 file path=opt/onbld/gk/gen_make.machines mode=0755
144 file path=opt/onbld/lib/$(ARCH)/libdwarf.so.1
145 file path=opt/onbld/lib/perl/onbld_elfmod.pm
146 file path=opt/onbld/lib/perl/onbld_elfmod_vertype.pm
151 file path=opt/onbld/lib/python2.4/onbld/Checks/CStyle.py mode=0444
152 file path=opt/onbld/lib/python2.4/onbld/Checks/CStyle.pyc mode=0444
153 file path=opt/onbld/lib/python2.4/onbld/Checks/Cddl.py mode=0444
154 file path=opt/onbld/lib/python2.4/onbld/Checks/Cddl.pyc mode=0444
155 file path=opt/onbld/lib/python2.4/onbld/Checks/CmtBlk.py mode=0444
156 file path=opt/onbld/lib/python2.4/onbld/Checks/CmtBlk.pyc mode=0444
157 file path=opt/onbld/lib/python2.4/onbld/Checks/Comments.py mode=0444
158 file path=opt/onbld/lib/python2.4/onbld/Checks/Comments.pyc mode=0444
159 file path=opt/onbld/lib/python2.4/onbld/Checks/Copyright.py mode=0444
160 file path=opt/onbld/lib/python2.4/onbld/Checks/Copyright.pyc mode=0444
161 file path=opt/onbld/lib/python2.4/onbld/Checks/DbLookups.py mode=0444
162 file path=opt/onbld/lib/python2.4/onbld/Checks/DbLookups.pyc mode=0444
163 file path=opt/onbld/lib/python2.4/onbld/Checks/HdrChk.py mode=0444
164 file path=opt/onbld/lib/python2.4/onbld/Checks/HdrChk.pyc mode=0444
165 file path=opt/onbld/lib/python2.4/onbld/Checks/JStyle.py mode=0444
166 file path=opt/onbld/lib/python2.4/onbld/Checks/JStyle.pyc mode=0444
167 file path=opt/onbld/lib/python2.4/onbld/Checks/Keywords.py mode=0444
168 file path=opt/onbld/lib/python2.4/onbld/Checks/Keywords.pyc mode=0444
169 file path=opt/onbld/lib/python2.4/onbld/Checks/Mapfile.py mode=0444
170 file path=opt/onbld/lib/python2.4/onbld/Checks/Mapfile.pyc mode=0444
171 file path=opt/onbld/lib/python2.4/onbld/Checks/ProcessCheck.py mode=0444
172 file path=opt/onbld/lib/python2.4/onbld/Checks/ProcessCheck.pyc mode=0444
173 file path=opt/onbld/lib/python2.4/onbld/Checks/__init__.py mode=0444
174 file path=opt/onbld/lib/python2.4/onbld/Checks/__init__.pyc mode=0444
175 file path=opt/onbld/lib/python2.4/onbld/Scm/Backup.py mode=0444
176 file path=opt/onbld/lib/python2.4/onbld/Scm/Backup.pyc mode=0444
177 file path=opt/onbld/lib/python2.4/onbld/Scm/Version.py mode=0444
178 file path=opt/onbld/lib/python2.4/onbld/Scm/Version.pyc mode=0444
179 file path=opt/onbld/lib/python2.4/onbld/Scm/WorkSpace.py mode=0444
180 file path=opt/onbld/lib/python2.4/onbld/Scm/WorkSpace.pyc mode=0444
181 file path=opt/onbld/lib/python2.4/onbld/Scm/__init__.py mode=0444
182 file path=opt/onbld/lib/python2.4/onbld/Scm/__init__.pyc mode=0444
183 file path=opt/onbld/lib/python2.4/onbld/__init__.py mode=0444
184 file path=opt/onbld/lib/python2.4/onbld/__init__.pyc mode=0444
185 file path=opt/onbld/lib/python2.4/onbld/hgext/__init__.py mode=0444
186 file path=opt/onbld/lib/python2.4/onbld/hgext/__init__.pyc mode=0444
187 file path=opt/onbld/lib/python2.4/onbld/hgext/cdm.py mode=0444
147 file path=opt/onbld/lib/python2.6/onbld/Checks/CStyle.py mode=0444
148 file path=opt/onbld/lib/python2.6/onbld/Checks/CStyle.pyc mode=0444
149 file path=opt/onbld/lib/python2.6/onbld/Checks/Cddl.py mode=0444
150 file path=opt/onbld/lib/python2.6/onbld/Checks/Cddl.pyc mode=0444
151 file path=opt/onbld/lib/python2.6/onbld/Checks/CmtBlk.py mode=0444
```



```
152 file path=opt/onbld/lib/python2.6/onbld/Checks/CmtBlk.pyc mode=0444
153 file path=opt/onbld/lib/python2.6/onbld/Checks/Comments.py mode=0444
154 file path=opt/onbld/lib/python2.6/onbld/Checks/Comments.pyc mode=0444
155 file path=opt/onbld/lib/python2.6/onbld/Checks/Copyright.py mode=0444
156 file path=opt/onbld/lib/python2.6/onbld/Checks/Copyright.pyc mode=0444
157 file path=opt/onbld/lib/python2.6/onbld/Checks/DbLookups.py mode=0444
158 file path=opt/onbld/lib/python2.6/onbld/Checks/DbLookups.pyc mode=0444
159 file path=opt/onbld/lib/python2.6/onbld/Checks/HdrChk.py mode=0444
160 file path=opt/onbld/lib/python2.6/onbld/Checks/HdrChk.pyc mode=0444
161 file path=opt/onbld/lib/python2.6/onbld/Checks/JStyle.py mode=0444
162 file path=opt/onbld/lib/python2.6/onbld/Checks/JStyle.pyc mode=0444
163 file path=opt/onbld/lib/python2.6/onbld/Checks/Keywords.py mode=0444
164 file path=opt/onbld/lib/python2.6/onbld/Checks/Keywords.pyc mode=0444
165 file path=opt/onbld/lib/python2.6/onbld/Checks/Mapfile.py mode=0444
166 file path=opt/onbld/lib/python2.6/onbld/Checks/Mapfile.pyc mode=0444
167 file path=opt/onbld/lib/python2.6/onbld/Checks/ProcessCheck.py mode=0444
168 file path=opt/onbld/lib/python2.6/onbld/Checks/ProcessCheck.pyc mode=0444
169 file path=opt/onbld/lib/python2.6/onbld/Checks/__init__.py mode=0444
170 file path=opt/onbld/lib/python2.6/onbld/Checks/__init__.pyc mode=0444
171 file path=opt/onbld/lib/python2.6/onbld/Scm/Backup.py mode=0444
172 file path=opt/onbld/lib/python2.6/onbld/Scm/Backup.pyc mode=0444
173 file path=opt/onbld/lib/python2.6/onbld/Scm/Version.py mode=0444
174 file path=opt/onbld/lib/python2.6/onbld/Scm/Version.pyc mode=0444
175 file path=opt/onbld/lib/python2.6/onbld/Scm/WorkSpace.py mode=0444
176 file path=opt/onbld/lib/python2.6/onbld/Scm/WorkSpace.pyc mode=0444
177 file path=opt/onbld/lib/python2.6/onbld/Scm/__init__.py mode=0444
178 file path=opt/onbld/lib/python2.6/onbld/Scm/__init__.pyc mode=0444
179 file path=opt/onbld/lib/python2.6/onbld/__init__.py mode=0444
180 file path=opt/onbld/lib/python2.6/onbld/__init__.pyc mode=0444
181 file path=opt/onbld/lib/python2.6/onbld/hgext/__init__.py mode=0444
182 file path=opt/onbld/lib/python2.6/onbld/hgext/__init__.pyc mode=0444
183 file path=opt/onbld/lib/python2.6/onbld/hgext/cdm.py mode=0444
184 file path=opt/onbld/man/man1/Install.1
185 file path=opt/onbld/man/man1/bldenv.1
186 file path=opt/onbld/man/man1/bringovercheck.1
187 file path=opt/onbld/man/man1/cddlchk.1
188 file path=opt/onbld/man/man1/check_rtime.1
189 file path=opt/onbld/man/man1/checkpaths.1
190 file path=opt/onbld/man/man1/codereview.1
191 file path=opt/onbld/man/man1/cstyle.1
192 file path=opt/onbld/man/man1/cw.1
193 file path=opt/onbld/man/man1/find_elf.1
194 file path=opt/onbld/man/man1/findunref.1
195 file path=opt/onbld/man/man1/flg.flp.1
196 file path=opt/onbld/man/man1/get_depend_info.1
197 file path=opt/onbld/man/man1/git-pbchk.1
198 file path=opt/onbld/man/man1/hdrchk.1
199 file path=opt/onbld/man/man1/hgsetup.1
200 file path=opt/onbld/man/man1/interface_check.1
201 file path=opt/onbld/man/man1/interface_cmp.1
202 file path=opt/onbld/man/man1/jstyle.1
203 file path=opt/onbld/man/man1/lintdump.1
204 file path=opt/onbld/man/man1/make_pkg_db.1
205 file path=opt/onbld/man/man1/mapfilechk.1
206 file path=opt/onbld/man/man1/ndrgen.1
207 file path=opt/onbld/man/man1/nightly.1
208 file path=opt/onbld/man/man1/onu.1
209 file path=opt/onbld/man/man1/sccscheck.1
210 file path=opt/onbld/man/man1/signit.1
211 file path=opt/onbld/man/man1/signproto.1
212 file path=opt/onbld/man/man1/webrev.1
213 file path=opt/onbld/man/man1/which_scm.1
214 file path=opt/onbld/man/man1/ws.1
215 file path=opt/onbld/man/man1/wsdiff.1
216 file path=opt/onbld/man/man1/xref.1
217 hardlink path=opt/onbld/bin/$(ARCH)/install.bin target=./install
```

```
218 legacy pkg=SUNWonbld desc="tools used to build the OS-Net consolidation" \
219     name="OS-Net Build Tools" version=11.11,REV=2009.10.22
220 license cr_Sun license=cr_Sun
221 license lic_CDDL license=lic_CDDL
222 license usr/src/tools/ctf/dwarf/THIRDPARTYLICENSE \
223     license=usr/src/tools/ctf/dwarf/THIRDPARTYLICENSE
224 license usr/src/tools/onbld/THIRDPARTYLICENSE \
225     license=usr/src/tools/onbld/THIRDPARTYLICENSE
226 link path=opt/onbld/bin/git-nits target=git-pbchk
227 link path=opt/onbld/lib/python target=python2.6
268 link path=opt/onbld/lib/python target=python2.4
228 link path=opt/onbld/man/man1/git-nits.1 target=git-pbchk.1
270 # DbLookups.py requires elementtree
271 depend fmri=library/python-2/python-extra-24 type=require
229 # webrev(1) requires ps2pdf
230 depend fmri=print/filter/ghostscript type=require
231 # hgsetup(1) uses check-hostname(1) and nightly sendmail(1M)
232 depend fmri=service/network/smtp/sendmail type=require
233 # nightly(1) uses wget
234 depend fmri=web/wget type=require
```

```
**********************************************************
    2848 Tue Sep 11 12:46:16 2012
new/usr/src/tools/Makefile
*** NO COMMENTS ***
**********************************************************
```

```
   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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
  24 #

  26 include ../Makefile.master

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

  31 BOOT_SUBDIRS= \
  32         cw

  34 COMMON_SUBDIRS= \
  35         codereview \
  36         codesign \
  37         cscope-fast \
  38         ctf \
  39         depcheck \
  40         env \
  41         findunref \
  42         ndrgen \
  43         onbld \
  44         pmodes \
  45         gk \
  46         install.bin \
  47         lintdump \
  48         protocmp \
  49         protolist \
  50         scripts

  52 #
  53 #  special versions of commands for use only in build
  54 #
  55 UNSHIPPED_SUBDIRS = \
  56         elfsign

  58 sparc_SUBDIRS= \
  59         chk4ubin \
  60         stabs \
  61         tokenize
```



```
  63 i386_SUBDIRS=          \
  64         aw              \
  65         elfextract      \
  66         mbh_patch

  68 LINTSUBDIRS= \
  69         codereview \
  70         ctf \
  71         cw \
  72         findunref \
  73         lintdump \
  74         ndrgen \
  75         protocmp \
  76         protolist

  78 SUBDIRS= \
  79         $($(MACH)_SUBDIRS) \
  80         $(COMMON_SUBDIRS) \
  81         $(UNSHIPPED_SUBDIRS)

  83 include Makefile.tools

  85 ROOTDIRS= \
  86         $(ROOTOPT) \
  87         $(ROOTONBLD) \
  88         $(ROOTONBLD)/bin \
  89         $(ROOTONBLD)/bin/$(MACH) \
  90         $(ROOTONBLD)/lib \
  91         $(ROOTONBLD)/lib/$(MACH) \
  92         $(ROOTONBLD)/lib/perl \
  93         $(ROOTONBLD)/lib/python2.4 \
  94         $(ROOTONBLD)/lib/python2.4/onbld \
  95         $(ROOTONBLD)/lib/python2.4/onbld/Checks \
  96         $(ROOTONBLD)/lib/python2.4/onbld/hgext \
  97         $(ROOTONBLD)/lib/python2.4/onbld/Scm \
  93         $(ROOTONBLD)/lib/python2.6 \
  94         $(ROOTONBLD)/lib/python2.6/onbld \
  95         $(ROOTONBLD)/lib/python2.6/onbld/Checks \
  96         $(ROOTONBLD)/lib/python2.6/onbld/hgext \
  97         $(ROOTONBLD)/lib/python2.6/onbld/Scm \
  98         $(ROOTONBLD)/env \
  99         $(ROOTONBLD)/etc \
 100         $(ROOTONBLD)/etc/exception_lists \
 101         $(ROOTONBLD)/gk \
 102         $(ROOTONBLD)/man \
 103         $(ROOTONBLD)/man/man1

 105 all :=          TARGET= install
 106 install :=      TARGET= install
 107 clean :=        TARGET= clean
 108 clobber :=      TARGET= clobber
 109 lint :=         TARGET= lint
 110 _msg :=         TARGET= _msg

 112 .KEEP_STATE:

 114 #
 115 # Only create directories in the tools proto area when doing an actual
 116 # build, not a clean or clobber.
 117 #
 118 DOROOTDIRS= $(ROOTDIRS)
 119 clobber:= DOROOTDIRS=
 120 clean:= DOROOTDIRS=

 122 all install: $(SUBDIRS)
```

```
124 clean: $(SUBDIRS)

126 clobber: $(SUBDIRS)
127         $(RM) -rf $(TOOLS_PROTO)

129 lint: $(LINTSUBDIRS)

131 _msg: $(MSGSUBDIRS)

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

135 $(SUBDIRS) $(CLOSED_SUBDIRS): $(BOOT_SUBDIRS)

137 $(BOOT_SUBDIRS) $(SUBDIRS): $$(DOROOTDIRS) $(ROOTONBLDLIBPY) FRC
138         @cd $@; pwd; $(MAKE) $(TARGET)

140 $(ROOTDIRS):
141         $(INS.dir)

143 $(ROOTONBLDLIBPY): $(ROOTDIRS)
144         $(RM) -r $@; $(SYMLINK) python2.6 $@
149         $(RM) -r $@; $(SYMLINK) python2.4 $@

146 FRC:
```

```
**********************************************************
    2846 Tue Sep 11 12:46:16 2012
new/usr/src/tools/Makefile.python
*** NO COMMENTS ***
**********************************************************
```

```
#
# 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.
#


#
# This Makefile provides a framework for building the onbld python
# modules with multiple versions of python.
#
# It expects as input:
#
#    PYSRCS   - List of python source files, these are also delivered as
#               build product.
#
#    PYOBJS   - List of compiled python (.pyc) files, with no directory prefix
#
#    PYTOPDIR - Absolute (including $(ROOT)) path to which files will
#               be installed, up until the version specific component.
#
#    PYMODDIR - Relative path to which files will be installed, below
#               the version specific component.
#
#    For example, to install to /opt/onbld/lib/onbld/python*/bar/
#
#          PYTOPDIR = $(ROOTONBLDLIB)
#          PYMODDIR = bar
#
#
# It provides as output:
#
#    ROOTPYFILES - The list of $(ROOT)-relative paths to which python
#                  source and binary files will be installed.  Your
#                  Makefile's 'install' target should depend upon
#                  this.
#
#    PYVERSOBJS  - The list of paths to compiled python build products,
#                  including their subdirectory.
#
#    pyclobber   - A target on which 'clobber' should depend, which
#                  removes the per-version python directories and the
#                  output within them.
```



```
#

PYFILES = $(PYSRCS) $(PYOBJS)

ROOTPYDIR_24 = $(PYTOPDIR)/python2.4/$(PYMODDIR)
ROOTPYFILES_24 = $(PYFILES:%=$(ROOTPYDIR_24)/%)

ROOTPYDIR_26 = $(PYTOPDIR)/python2.6/$(PYMODDIR)
ROOTPYFILES_26  = $(PYFILES:%=$(ROOTPYDIR_26)/%)

ROOTPYFILES = $(ROOTPYFILES_26)
ROOTPYFILES = $(ROOTPYFILES_24) $(ROOTPYFILES_26)
$(ROOTPYFILES) := FILEMODE = 0444

PYVERSDIRS = python2.6
PYVERSDIRS = python2.4 python2.6

PY24OBJS = $(PYOBJS:%=python2.4/%)
$(PY24OBJS) := PYTHON = $(PYTHON_24)

PY26OBJS = $(PYOBJS:%=python2.6/%)
$(PY26OBJS) := PYTHON = $(PYTHON_26)

PYVERSOBJS = $(PY26OBJS)
PYVERSOBJS = $(PY24OBJS) $(PY26OBJS)

CLOBBERFILES += $(PYVERSOBJS)
CLOBBERDIRS += $(PYVERSDIRS)

.KEEP_STATE:

python2.6/%.pyc: %.py
python2.4/%.pyc python2.6/%.pyc: %.py
	@[ -d $(@D) ] || mkdir $(@D)
	$(RM) $@
	$(PYTHON) -mpy_compile $<
	$(MV) $(*).pyc $@

$(ROOTPYDIR_24)/%.pyc: python2.4/%.pyc
	$(INS.pyfile)

$(ROOTPYDIR_26)/%.pyc: python2.6/%.pyc
	$(INS.pyfile)

$(ROOTPYDIR_26)/%.py: %.py
$(ROOTPYDIR_24)/%.py $(ROOTPYDIR_26)/%.py: %.py
	$(INS.pyfile)

pyclobber:
	$(RM) $(CLOBBERFILES)
	$(RM) -rf $(CLOBBERDIRS)
```

```
**********************************************************
    3115 Tue Sep 11 12:46:17 2012
new/usr/src/tools/Makefile.tools
*** NO COMMENTS ***
**********************************************************
```

```
#
# 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)
ROOTONBLDLIBPERL=       $(ROOTONBLD)/lib/perl
ROOTONBLDLIBPY=         $(ROOTONBLD)/lib/python
ROOTONBLDLIBPY_24=      $(ROOTONBLD)/lib/python2.4
ROOTONBLDLIBPY_26=      $(ROOTONBLD)/lib/python2.6
ROOTONBLDENV=           $(ROOTONBLD)/env
ROOTONBLDGK=            $(ROOTONBLD)/gk
ROOTONBLDMAN=           $(ROOTONBLD)/man
ROOTONBLDMAN1=          $(ROOTONBLD)/man/man1
ROOTONBLDETCABI=        $(ROOTONBLD)/etc/abi
ROOTONBLDETCEXCEPT=     $(ROOTONBLD)/etc/exception_lists

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)/%)
ROOTONBLDMAN1FILES=     $(MAN1FILES:%=$(ROOTONBLDMAN1)/%)
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)

$(ROOTONBLDMAN1)/%: %
        $(INS.file)

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

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

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

```
**********************************************************
    3704 Tue Sep 11 12:46:18 2012
new/usr/src/tools/scripts/cddlchk.py
*** NO COMMENTS ***
**********************************************************
```

```
  1 #!/usr/bin/python2.6
  1 #!/usr/bin/python2.4
  2 #
  3 # CDDL HEADER START
  4 #
  5 # The contents of this file are subject to the terms of the
  6 # Common Development and Distribution License (the "License").
  7 # You may not use this file except in compliance with the License.
  8 #
  9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 10 # or http://www.opensolaris.org/os/licensing.
 11 # See the License for the specific language governing permissions
 12 # and limitations under the License.
 13 #
 14 # When distributing Covered Code, include this CDDL HEADER in each
 15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 16 # If applicable, add the following below this CDDL HEADER, with the
 17 # fields enclosed by brackets "[]" replaced with your own identifying
 18 # information: Portions Copyright [yyyy] [name of copyright owner]
 19 #
 20 # CDDL HEADER END
 21 #

 23 #
 24 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
 25 #

 27 #
 28 # Check for valid CDDL blocks in source files.
 29 #

 31 import sys, os, getopt, fnmatch

 33 sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
 34                                 "python%d.%d" % sys.version_info[:2]))

 36 # Allow running from the source tree, using the modules in the source tree
 37 sys.path.insert(2, os.path.join(os.path.dirname(__file__), '..'))

 39 from onbld.Checks.Cddl import cddlchk

 41 class ExceptionList(object):
 42         def __init__(self):
 43                 self.dirs = []
 44                 self.files = []
 45                 self.extensions = []

 47         def load(self, exfile):
 48                 fh = None
 49                 try:
 50                         fh = open(exfile, 'r')
 51                 except IOError, e:
 52                         sys.stderr.write('Failed to open exception list: '
 53                                          '%s: %s\n' % (e.filename, e.strerror))
 54                         sys.exit(2)

 56                 for line in fh:
 57                         line = line.strip()

 59                         if line.strip().endswith('/'):
 60                                 self.dirs.append(line[0:-1])
```



```
 61                         elif line.startswith('*.'):
 62                                 self.extensions.append(line)
 63                         else:
 64                                 self.files.append(line)

 66                 fh.close()

 68         def match(self, filename):
 69                 if os.path.isdir(filename):
 70                         return filename in self.dirs
 71                 else:
 72                         if filename in self.files:
 73                                 return True

 75                         for pat in self.extensions:
 76                                 if fnmatch.fnmatch(filename, pat):
 77                                         return True

 79         def __contains__(self, elt):
 80                 return self.match(elt)

 82 def usage():
 83         progname = os.path.split(sys.argv[0])[1]
 84         sys.stderr.write('''Usage: %s [-av] [-x exceptions] paths...
 85         -a              check that all the specified files have a CDDL block.
 86         -v              report on all files, not just those with errors.
 87         -x exceptions   load an exceptions file
 88 ''' % progname)
 89         sys.exit(2)


 92 def check(filename, opts):
 93         try:
 94                 fh = open(filename, 'r')
 95         except IOError, e:
 96                 sys.stderr.write("failed to open '%s': %s\n" %
 97                                  (e.filename, e.strerror))
 98                 return 1
 99         else:
100                 return cddlchk(fh, verbose=opts['verbose'],
101                                lenient=opts['lenient'],
102                                output=sys.stdout)

104 def walker(opts, dirname, fnames):
105         for f in fnames:
106                 path = os.path.join(dirname, f)

108                 if not os.path.isdir(path):
109                         if not path in opts['exclude']:
110                                 opts['status'] |= check(path, opts)
111                 else:
112                         if path in opts['exclude']:
113                                 fnames.remove(f)

115 def walkpath(path, opts):
116         if os.path.isdir(path):
117                 os.path.walk(path, walker, opts)
118         else:
119                 if not path in opts['exclude']:
120                         opts['status'] |= check(path, opts)

122 def main(args):
123         options = {
124                 'status': 0,
125                 'lenient': True,
126                 'verbose': False,
```

```
127                 'exclude': ExceptionList()
128         }

130         try:
131                 opts, args = getopt.getopt(sys.argv[1:], 'avx:')
132         except getopt.GetoptError:
133                 usage()
134                 sys.exit(2)

136         for opt, arg in opts:
137                 if opt == '-a':
138                         options['lenient'] = False
139                 elif opt == '-v':
140                         options['verbose'] = True
141                 elif opt == '-x':
142                         options['exclude'].load(arg)

144         for path in args:
145                 walkpath(path, options)

147         return options['status']

149 if __name__ == '__main__':
150         sys.exit(main(sys.argv[1:]))
```

```
**********************************************************
    1569 Tue Sep 11 12:46:19 2012
new/usr/src/tools/scripts/copyrightchk.py
*** NO COMMENTS ***
**********************************************************
```

```
  1 #!/usr/bin/python2.6
  1 #!/usr/bin/python2.4
  2 #
  3 # CDDL HEADER START
  4 #
  5 # The contents of this file are subject to the terms of the
  6 # Common Development and Distribution License (the "License").
  7 # You may not use this file except in compliance with the License.
  8 #
  9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 10 # or http://www.opensolaris.org/os/licensing.
 11 # See the License for the specific language governing permissions
 12 # and limitations under the License.
 13 #
 14 # When distributing Covered Code, include this CDDL HEADER in each
 15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 16 # If applicable, add the following below this CDDL HEADER, with the
 17 # fields enclosed by brackets "[]" replaced with your own identifying
 18 # information: Portions Copyright [yyyy] [name of copyright owner]
 19 #
 20 # CDDL HEADER END
 21 #

 23 #
 24 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
 25 #

 27 #
 28 # Check for valid SMI copyright notices in source files.
 29 #

 31 import sys, os

 33 sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
 34                                 "python%d.%d" % sys.version_info[:2]))

 36 # Allow running from the source tree, using the modules in the source tree
 37 sys.path.insert(2, os.path.join(os.path.dirname(__file__), '..'))

 39 from onbld.Checks.Copyright import copyright

 41 ret = 0
 42 for filename in sys.argv[1:]:
 43         try:
 44                 fin = open(filename, 'r')
 45         except IOError, e:
 46                 sys.stderr.write("failed to open '%s': %s\n" %
 47                                  (e.filename, e.strerror))
 48                 continue

 50         ret |= copyright(fin, output=sys.stdout)
 51         fin.close()

 53 sys.exit(ret)
```

```
**********************************************************
   10548 Tue Sep 11 12:46:19 2012
new/usr/src/tools/scripts/git-pbchk.py
*** NO COMMENTS ***
**********************************************************
```

```
#!/usr/bin/python2.6
#!/usr/bin/python2.4
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License version 2
#  as published by the Free Software Foundation.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

#
# Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2008, 2012 Richard Lowe
#

import getopt
import os
import re
import subprocess
import sys
import tempfile

from cStringIO import StringIO

# This is necessary because, in a fit of pique, we used hg-format ignore lists
# for NOT files.
from mercurial import ignore

#
# Adjust the load path based on our location and the version of python into
# which it is being loaded.  This assumes the normal onbld directory
# structure, where we are in bin/ and the modules are in
# lib/python(version)?/onbld/Scm/.  If that changes so too must this.
#
sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
                                "python%d.%d" % sys.version_info[:2]))

#
# Add the relative path to usr/src/tools to the load path, such that when run
# from the source tree we use the modules also within the source tree.
#
sys.path.insert(2, os.path.join(os.path.dirname(__file__), ".."))

from onbld.Checks import Comments, Copyright, CStyle, HdrChk
from onbld.Checks import JStyle, Keywords, Mapfile


class GitError(Exception):
    pass

def git(command):
    """Run a command and return a stream containing its stdout (and write its
    stderr to its stdout)"""
```



```
    if type(command) != list:
        command = command.split()

    command = ["git"] + command

    try:
        tmpfile = tempfile.TemporaryFile(prefix="git-nits")
    except EnvironmentError, e:
        raise GitError("Could not create temporary file: %s\n" % e)

    try:
        p = subprocess.Popen(command,
                             stdout=tmpfile,
                             stderr=subprocess.STDOUT)
    except OSError, e:
        raise GitError("could not execute %s: %s\n" (command, e))

    err = p.wait()
    if err != 0:
        raise GitError(p.stdout.read())

    tmpfile.seek(0)
    return tmpfile


def git_root():
    """Return the root of the current git workspace"""

    p = git('rev-parse --git-dir')

    if not p:
        sys.stderr.write("Failed finding git workspace\n")
        sys.exit(err)

    return os.path.abspath(os.path.join(p.readlines()[0],
                                        os.path.pardir))


def git_branch():
    """Return the current git branch"""

    p = git('branch')

    if not p:
        sys.stderr.write("Failed finding git branch\n")
        sys.exit(err)

    for elt in p:
        if elt[0] == '*':
            if elt.endswith('(no branch)'):
                return None
            return elt.split()[1]


def git_parent_branch(branch):
    """Return the parent of the current git branch.

    If this branch tracks a remote branch, return the remote branch which is
    tracked.  If not, default to origin/master."""

    if not branch:
        return None

    p = git("for-each-ref --format=%(refname:short) %(upstream:short) " +
            "refs/heads/")
```

```
127     if not p:
128         sys.stderr.write("Failed finding git parent branch\n")
129         sys.exit(err)
130
131     for line in p:
132         # Git 1.7 will leave a ' ' trailing any non-tracking branch
133         if ' ' in line and not line.endswith(' \n'):
134             local, remote = line.split()
135             if local == branch:
136                 return remote
137     return 'origin/master'
138
139
140 def git_comments(parent):
141     """Return a list of any checkin comments on this git branch"""
142
143     p = git('log --pretty=format:%%B %s..' % parent)
144
145     if not p:
146         sys.stderr.write("Failed getting git comments\n")
147         sys.exit(err)
148
149     return map(lambda x: x.strip(), p.readlines())
150
151
152 def git_file_list(parent, paths=None):
153     """Return the set of files which have ever changed on this branch.
154
155     NB: This includes files which no longer exist, or no longer actually
156     differ."""
157
158     p = git("log --name-only --pretty=format: %s.. %s" %
159             (parent, ' '.join(paths)))
160
161     if not p:
162         sys.stderr.write("Failed building file-list from git\n")
163         sys.exit(err)
164
165     ret = set()
166     for fname in p:
167         if fname and not fname.isspace() and fname not in ret:
168             ret.add(fname.strip())
169
170     return ret
171
172
173 def not_check(root, cmd):
174     """Return a function which returns True if a file given as an argument
175     should be excluded from the check named by 'cmd'"""
176
177     ignorefiles = filter(os.path.exists,
178                          [os.path.join(root, ".git", "%s.NOT" % cmd),
179                           os.path.join(root, "exception_lists", cmd)])
180     if len(ignorefiles) > 0:
181         return ignore.ignore(root, ignorefiles, sys.stderr.write)
182     else:
183         return lambda x: False
184
185
186 def gen_files(root, parent, paths, exclude):
187     """Return a function producing file names, relative to the current
188     directory, of any file changed on this branch (limited to 'paths' if
189     requested), and excluding files for which exclude returns a true value """
190
191     # Taken entirely from Python 2.6's os.path.relpath which we would use if we
192     # could.
```



```
193     def relpath(path, here):
194         c = os.path.abspath(os.path.join(root, path)).split(os.path.sep)
195         s = os.path.abspath(here).split(os.path.sep)
196         l = len(os.path.commonprefix((s, c)))
197         return os.path.join(*[os.path.pardir] * (len(s)-l) + c[l:])
198
199     def ret(select=None):
200         if not select:
201             select = lambda x: True
202
203         for f in git_file_list(parent, paths):
204             f = relpath(f, '.')
205             if (os.path.exists(f) and select(f) and not exclude(f)):
206                 yield f
207     return ret
208
209
210 def comchk(root, parent, flist, output):
211     output.write("Comments:\n")
212
213     return Comments.comchk(git_comments(parent), check_db=True,
214                            output=output)
215
216
217 def mapfilechk(root, parent, flist, output):
218     ret = 0
219
220     # We are interested in examining any file that has the following
221     # in its final path segment:
222     #    - Contains the word 'mapfile'
223     #    - Begins with 'map.'
224     #    - Ends with '.map'
225     # We don't want to match unless these things occur in final path segment
226     # because directory names with these strings don't indicate a mapfile.
227     # We also ignore files with suffixes that tell us that the files
228     # are not mapfiles.
229     MapfileRE = re.compile(r'.*((mapfile[^/]*)|(/map\.+[^/]*)|(\.map))$',
230         re.IGNORECASE)
231     NotMapSuffixRE = re.compile(r'.*\.[ch]$', re.IGNORECASE)
232
233     output.write("Mapfile comments:\n")
234
235     for f in flist(lambda x: MapfileRE.match(x) and not
236                    NotMapSuffixRE.match(x)):
237         fh = open(f, 'r')
238         ret |= Mapfile.mapfilechk(fh, output=output)
239         fh.close()
240     return ret
241
242
243 def copyright(root, parent, flist, output):
244     ret = 0
245     output.write("Copyrights:\n")
246     for f in flist():
247         fh = open(f, 'r')
248         ret |= Copyright.copyright(fh, output=output)
249         fh.close()
250     return ret
251
252
253 def hdrchk(root, parent, flist, output):
254     ret = 0
255     output.write("Header format:\n")
256     for f in flist(lambda x: x.endswith('.h')):
257         fh = open(f, 'r')
258         ret |= HdrChk.hdrchk(fh, lenient=True, output=output)
```

```
259         fh.close()
260     return ret


263 def cstyle(root, parent, flist, output):
264     ret = 0
265     output.write("C style:\n")
266     for f in flist(lambda x: x.endswith('.c') or x.endswith('.h')):
267         fh = open(f, 'r')
268         ret |= CStyle.cstyle(fh, output=output, picky=True,
269                              check_posix_types=True,
270                              check_continuation=True)
271         fh.close()
272     return ret


275 def jstyle(root, parent, flist, output):
276     ret = 0
277     output.write("Java style:\n")
278     for f in flist(lambda x: x.endswith('.java')):
279         fh = open(f, 'r')
280         ret |= JStyle.jstyle(fh, output=output, picky=True)
281         fh.close()
282     return ret


285 def keywords(root, parent, flist, output):
286     ret = 0
287     output.write("SCCS Keywords:\n")
288     for f in flist():
289         fh = open(f, 'r')
290         ret |= Keywords.keywords(fh, output=output)
291         fh.close()
292     return ret


295 def run_checks(root, parent, cmds, paths='', opts={}):
296     """Run the checks given in 'cmds', expected to have well-known signatures,
297     and report results for any which fail.

299     Return failure if any of them did.

301     NB: the function name of the commands passed in is used to name the NOT
302     file which excepts files from them."""

304     ret = 0

306     for cmd in cmds:
307         s = StringIO()

309         exclude = not_check(root, cmd.func_name)
310         result = cmd(root, parent, gen_files(root, parent, paths, exclude),
311                      output=s)
312         ret |= result

314         if result != 0:
315             print s.getvalue()

317     return ret


320 def nits(root, parent, paths):
321     cmds = [copyright,
322             cstyle,
323             hdrchk,
324             jstyle,
```



```
325             keywords,
326             mapfilechk]
327     run_checks(root, parent, cmds, paths)


330 def pbchk(root, parent, paths):
331     cmds = [comchk,
332             copyright,
333             cstyle,
334             hdrchk,
335             jstyle,
336             keywords,
337             mapfilechk]
338     run_checks(root, parent, cmds)


341 def main(cmd, args):
342     parent_branch = None

344     try:
345         opts, args = getopt.getopt(args, 'b:')
346     except getopt.GetoptError, e:
347         sys.stderr.write(str(e) + '\n')
348         sys.stderr.write("Usage: %s [-b branch] [path...]\n" % cmd)
349         sys.exit(1)

351     for opt, arg in opts:
352         if opt == '-b':
353             parent_branch = arg

355     if not parent_branch:
356         parent_branch = git_parent_branch(git_branch())

358     func = nits
359     if cmd == 'git-pbchk':
360         func = pbchk
361         if args:
362             sys.stderr.write("only complete workspaces may be pbchk'd\n");
363             sys.exit(1)

365     func(git_root(), parent_branch, args)

367 if __name__ == '__main__':
368     try:
369         main(os.path.basename(sys.argv[0]), sys.argv[1:])
370     except GitError, e:
371         sys.stderr.write("failed to run git:\n %s\n" % str(e))
372         sys.exit(1)
```

```
**********************************************************
    1933 Tue Sep 11 12:46:20 2012
new/usr/src/tools/scripts/hdrchk.py
*** NO COMMENTS ***
**********************************************************
```

```
  1 #!/usr/bin/python2.6
  1 #!/usr/bin/python2.4
  2 #
  3 # CDDL HEADER START
  4 #
  5 # The contents of this file are subject to the terms of the
  6 # Common Development and Distribution License (the "License").
  7 # You may not use this file except in compliance with the License.
  8 #
  9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 10 # or http://www.opensolaris.org/os/licensing.
 11 # See the License for the specific language governing permissions
 12 # and limitations under the License.
 13 #
 14 # When distributing Covered Code, include this CDDL HEADER in each
 15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 16 # If applicable, add the following below this CDDL HEADER, with the
 17 # fields enclosed by brackets "[]" replaced with your own identifying
 18 # information: Portions Copyright [yyyy] [name of copyright owner]
 19 #
 20 # CDDL HEADER END
 21 #

 23 #
 24 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
 25 #

 27 #
 28 # Check header files conform to ON standards.
 29 #

 31 import sys, os, getopt

 33 sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
 34                                 "python%d.%d" % sys.version_info[:2]))

 36 # Allow running from the source tree, using the modules in the source tree
 37 sys.path.insert(2, os.path.join(os.path.dirname(__file__), '..'))

 39 from onbld.Checks.HdrChk import hdrchk

 41 def usage():
 42         progname = os.path.split(sys.argv[0])[1]
 43         msg =  ['Usage: %s [-a] file [file...]\n' % progname,
 44                 '  -a\tApply (more lenient) application header rules\n']
 45         sys.stderr.writelines(msg)


 48 try:
 49         opts, args = getopt.getopt(sys.argv[1:], 'a')
 50 except getopt.GetoptError:
 51         usage()
 52         sys.exit(2)

 54 lenient = False
 55 for opt, arg in opts:
 56         if opt == '-a':
 57                 lenient = True

 59 ret = 0
 60 for filename in args:
```



```
 61         try:
 62                 fh = open(filename, 'r')
 63         except IOError, e:
 64                 sys.stderr.write("failed to open '%s': %s\n" %
 65                                  (e.filename, e.strerror))
 66         else:
 67                 ret |= hdrchk(fh, lenient=lenient, output=sys.stderr)
 68                 fh.close()
 69 sys.exit(ret)
```

```
**********************************************************
    3478 Tue Sep 11 12:46:21 2012
new/usr/src/tools/scripts/hg-active.py
*** NO COMMENTS ***
**********************************************************
```

```
  1 #!/usr/bin/python2.6
  1 #!/usr/bin/python2.4
  2 #
  3 #  This program is free software; you can redistribute it and/or modify
  4 #  it under the terms of the GNU General Public License version 2
  5 #  as published by the Free Software Foundation.
  6 #
  7 #  This program is distributed in the hope that it will be useful,
  8 #  but WITHOUT ANY WARRANTY; without even the implied warranty of
  9 #  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 10 #  GNU General Public License for more details.
 11 #
 12 #  You should have received a copy of the GNU General Public License
 13 #  along with this program; if not, write to the Free Software
 14 #  Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 15 #

 17 #
 18 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
 19 #

 21 '''
 22 Create a wx-style active list on stdout based on a Mercurial
 23 workspace in support of webrev's Mercurial support.
 24 '''

 26 #
 27 # NB: This assumes the normal onbld directory structure
 28 #
 29 import sys, os

 31 sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
 32                                 "python%d.%d" % sys.version_info[:2]))

 34 # Allow running from the source tree, using the modules in the source tree
 35 sys.path.insert(2, os.path.join(os.path.dirname(__file__), ".."))

 37 from onbld.Scm import Version

 39 try:
 40     Version.check_version()
 41 except Version.VersionMismatch, versionerror:
 42     sys.stderr.write("Error: %s\n" % versionerror)
 43     sys.exit(1)


 46 import getopt, binascii
 47 from mercurial import error, hg, ui, util
 48 from onbld.Scm.WorkSpace import WorkSpace


 51 def usage():
 52     sys.stderr.write("usage: %s [-p parent] -w workspace\n" %
 53                      os.path.basename(__file__))
 54     sys.exit(2)


 57 def main(argv):
 58     try:
 59         opts = getopt.getopt(argv, 'w:o:p:')[0]
 60     except getopt.GetoptError, e:
```



```
 61         sys.stderr.write(str(e) + '\n')
 62         usage()

 64     parentpath = None
 65     wspath = None
 66     outputfile = None

 68     for opt, arg in opts:
 69         if opt == '-w':
 70             wspath = arg
 71         elif opt == '-o':
 72             outputfile = arg
 73         elif opt == '-p':
 74             parentpath = arg

 76     if not wspath:
 77         usage()

 79     try:
 80         repository = hg.repository(ui.ui(), wspath)
 81     except error.RepoError, e:
 82         sys.stderr.write("failed to open repository: %s\n" % e)
 83         sys.exit(1)

 85     ws = WorkSpace(repository)
 86     act = ws.active(parentpath)

 88     node = act.parenttip.node()
 89     parenttip = binascii.hexlify(node)

 91     fh = None
 92     if outputfile:
 93         try:
 94             fh = open(outputfile, 'w')
 95         except EnvironmentError, e:
 96             sys.stderr.write("could not open output file: %s\n" % e)
 97             sys.exit(1)
 98     else:
 99         fh = sys.stdout

101     fh.write("HG_PARENT=%s\n" % parenttip)

103     entries = [i for i in act]
104     entries.sort()

106     for entry in entries:
107         if entry.is_renamed() or entry.is_copied():
108             fh.write("%s %s\n" % (entry.name, entry.parentname))
109         else:
110             fh.write("%s\n" % entry.name)

112         # Strip blank lines.
113         comments = filter(lambda x: x and not x.isspace(),
114                           entry.comments)

116         fh.write('\n')
117         if comments:
118             fh.write('%s\n' % '\n'.join(comments))
119         else:
120             fh.write("*** NO COMMENTS ***\n")
121         fh.write('\n')

123 if __name__ == '__main__':
124     try:
125         main(sys.argv[1:])
126     except KeyboardInterrupt:
```

```
 127         sys.exit(1)
 128     except util.Abort, msg:
 129         sys.stderr.write("Abort: %s\n" % msg)
 130         sys.exit(1)
```

```
**********************************************************
    3565 Tue Sep 11 12:46:22 2012
new/usr/src/tools/scripts/mapfilechk.py
*** NO COMMENTS ***
**********************************************************
```

```
#!/usr/bin/python2.6
#!/usr/bin/python2.4
#
# 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) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
#

#
# Check for valid link-editor mapfile comment blocks in source files.
#

import sys, os, getopt, fnmatch

sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
                                "python%d.%d" % sys.version_info[:2]))

# Allow running from the source tree, using the modules in the source tree
sys.path.insert(2, os.path.join(os.path.dirname(__file__), '..'))

from onbld.Checks.Mapfile import mapfilechk

class ExceptionList(object):
        def __init__(self):
                self.dirs = []
                self.files = []
                self.extensions = []

        def load(self, exfile):
                fh = None
                try:
                        fh = open(exfile, 'r')
                except IOError, e:
                        sys.stderr.write('Failed to open exception list: '
                                         '%s: %s\n' % (e.filename, e.strerror))
                        sys.exit(2)

                for line in fh:
                        line = line.strip()

                        if line.strip().endswith('/'):
                                self.dirs.append(line[0:-1])
```



```
                        elif line.startswith('*.'):
                                self.extensions.append(line)
                        else:
                                self.files.append(line)

                fh.close()

        def match(self, filename):
                if os.path.isdir(filename):
                        return filename in self.dirs
                else:
                        if filename in self.files:
                                return True

                        for pat in self.extensions:
                                if fnmatch.fnmatch(filename, pat):
                                        return True

        def __contains__(self, elt):
                return self.match(elt)

def usage():
        progname = os.path.split(sys.argv[0])[1]
        sys.stderr.write('''Usage: %s [-v] [-x exceptions] paths...
        -v              report on all files, not just those with errors.
        -x exceptions   load an exceptions file
''' % progname)
        sys.exit(2)


def check(filename, opts):
        try:
                fh = open(filename, 'r')
        except IOError, e:
                sys.stderr.write("failed to open '%s': %s\n" %
                                 (e.filename, e.strerror))
                return 1
        else:
                return mapfilechk(fh, verbose=opts['verbose'],
                                  output=sys.stdout)

def walker(opts, dirname, fnames):
        for f in fnames:
                path = os.path.join(dirname, f)

                if not os.path.isdir(path):
                        if not path in opts['exclude']:
                                opts['status'] |= check(path, opts)
                else:
                        if path in opts['exclude']:
                                fnames.remove(f)

def walkpath(path, opts):
        if os.path.isdir(path):
                os.path.walk(path, walker, opts)
        else:
                if not path in opts['exclude']:
                        opts['status'] |= check(path, opts)

def main(args):
        options = {
                'status': 0,
                'verbose': False,
                'exclude': ExceptionList()
        }
```

```
127         try:
128                 opts, args = getopt.getopt(sys.argv[1:], 'avx:')
129         except getopt.GetoptError:
130                 usage()
131                 sys.exit(2)

133         for opt, arg in opts:
134                 if opt == '-v':
135                         options['verbose'] = True
136                 elif opt == '-x':
137                         options['exclude'].load(arg)

139         for path in args:
140                 walkpath(path, options)

142         return options['status']

144 if __name__ == '__main__':
145         sys.exit(main(sys.argv[1:]))
```