```
**********************************************************
   35131 Wed Sep  2 16:30:34 2015
new/usr/src/Makefile.master
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
```

```
#
# 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.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2015 Gary Mills
#

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

#
# Adjunct root, containing an additional proto area to be used for headers
# and libraries.
#
ADJUNCT_PROTO=

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

#
# RELEASE_BUILD should be cleared for final release builds.
# NOT_RELEASE_BUILD is exactly what the name implies.
#
# __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=
RELEASE_BUILD=                          $(POUND_SIGN)
$(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
PATCH_BUILD=                            $(POUND_SIGN)

# 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=)

# 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.
# __GNUC and GCC 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.
#

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

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

# define runtime JAVA_HOME, primarily for cmd/pools/poold
JAVA_HOME=      /usr/java
# define buildtime JAVA_ROOT
JAVA_ROOT=      /usr/java

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

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

```
128 RPCGEN=         /usr/bin/rpcgen
129 STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
130 ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
131 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
132 ECHO=           echo
133 INS=            install
134 TRUE=           true
135 SYMLINK=        /usr/bin/ln -s
136 LN=             /usr/bin/ln
137 CHMOD=          /usr/bin/chmod
138 MV=             /usr/bin/mv -f
139 RM=             /usr/bin/rm -f
140 CUT=            /usr/bin/cut
141 NM=             /usr/ccs/bin/nm
142 DIFF=           /usr/bin/diff
143 GREP=           /usr/bin/grep
144 EGREP=          /usr/bin/egrep
145 ELFWRAP=        /usr/bin/elfwrap
146 KSH93=          /usr/bin/ksh93
147 SED=            /usr/bin/sed
148 NAWK=           /usr/bin/nawk
149 CP=             /usr/bin/cp -f
150 MCS=            /usr/ccs/bin/mcs
151 CAT=            /usr/bin/cat
152 ELFDUMP=        /usr/ccs/bin/elfdump
153 M4=             /usr/bin/m4
152 M4=             /usr/ccs/bin/m4
154 STRIP=          /usr/ccs/bin/strip
155 LEX=            /usr/ccs/bin/lex
156 FLEX=           /usr/bin/flex
157 YACC=           /usr/ccs/bin/yacc
158 CPP=            /usr/lib/cpp
159 JAVAC=          $(JAVA_ROOT)/bin/javac
160 JAVAH=          $(JAVA_ROOT)/bin/javah
161 JAVADOC=        $(JAVA_ROOT)/bin/javadoc
162 RMIC=           $(JAVA_ROOT)/bin/rmic
163 JAR=            $(JAVA_ROOT)/bin/jar
164 CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
165 CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
166 CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
167 CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
168 NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
169 GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
170 XREF=           $(ONBLD_TOOLS)/bin/xref
171 FIND=           /usr/bin/find
172 PERL=           /usr/bin/perl
173 PERL_VERSION=   5.10.0
174 PERL_PKGVERS=   -510
175 PERL_ARCH =             i86pc-solaris-64int
176 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
177 PYTHON_26=      /usr/bin/python2.6
178 PYTHON=         $(PYTHON_26)
179 SORT=           /usr/bin/sort
180 TOUCH=          /usr/bin/touch
181 WC=             /usr/bin/wc
182 XARGS=          /usr/bin/xargs
183 ELFEDIT=        /usr/bin/elfedit
184 ELFSIGN=        /usr/bin/elfsign
185 DTRACE=         /usr/sbin/dtrace -xnolibs
186 UNIQ=           /usr/bin/uniq
187 TAR=            /usr/bin/tar
188 ASTBINDIR=      /usr/ast/bin
189 MSGCC=          $(ASTBINDIR)/msgcc

191 FILEMODE=       644
192 DIRMODE=        755
```



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

198 # For stylistic checks
199 #
200 # Note that the X and C checks are not used at this time and may need
201 # modification when they are actually used.
202 #
203 CSTYLE=         $(ONBLD_TOOLS)/bin/cstyle
204 CSTYLE_TAIL=
205 HDRCHK=         $(ONBLD_TOOLS)/bin/hdrchk
206 HDRCHK_TAIL=
207 JSTYLE=         $(ONBLD_TOOLS)/bin/jstyle

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

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

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

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

227 INS.file=       $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
228 INS.dir=        $(INS) -s -d -m $(DIRMODE) $@
229 # installs and renames at once
230 #
231 INS.rename=     $(INS.file); $(MV) $(@D)/$(<F) $@

233 # install a link
234 INSLINKTARGET=  $<
235 INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@
236 INS.symlink=    $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

238 #
239 # Python bakes the mtime of the .py file into the compiled .pyc and
240 # rebuilds if the baked-in mtime != the mtime of the source file
241 # (rather than only if it's less than), thus when installing python
242 # files we must make certain to not adjust the mtime of the source
243 # (.py) file.
244 #
245 INS.pyfile=     $(INS.file); $(TOUCH) -r $< $@

247 # MACH must be set in the shell environment per uname -p on the build host
248 # More specific architecture variables should be set in lower makefiles.
249 #
250 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
251 # architectures on which we do not build 64-bit versions.
252 # (There are no such architectures at the moment.)
253 #
254 # Set BUILD64=# in the environment to disable 64-bit amd64
255 # builds on i386 machines.

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

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

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

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

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

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

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

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

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

# One optimization the compiler might perform is to turn this:
#       #pragma weak foo
#       extern int foo;
#       if (&foo)
#               foo = 5;
# into
#       foo = 5;
# Since we do some of this (foo might be referenced in common kernel code
```



```
# but provided only for some cpu modules or platforms), we disable this
# optimization.
#
sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
i386_CCUNBOUND  =
CCUNBOUND       = $($(MACH)_CCUNBOUND)

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

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

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

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

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

#
# disable the incremental linker
ILDOFF=                 -xildoff
#
XDEPEND=                -xdepend
XFFLAG=                 -xF=%all
XESS=                   -xs
XSTRCONST=              -xstrconst

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

CERRWARN += -_gcc=-Wno-missing-braces
CERRWARN += -_gcc=-Wno-sign-compare
CERRWARN += -_gcc=-Wno-unknown-pragmas
CERRWARN += -_gcc=-Wno-unused-parameter
CERRWARN += -_gcc=-Wno-missing-field-initializers

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

```
391 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
392 # -nd builds
393 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
394 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

396 #
397 # turn warnings into errors (C++)
398 CCERRWARN=              -xwe

400 # C99 mode
401 C99_ENABLE=     -xc99=%all
402 C99_DISABLE=    -xc99=%none
403 C99MODE=        $(C99_DISABLE)
404 C99LMODE=       $(C99MODE:-xc99%=-Xc99%)

406 # In most places, assignments to these macros should be appended with +=
407 # (CPPFLAGS.first allows values to be prepended to CPPFLAGS).
408 sparc_CFLAGS=   $(sparc_XARCH) $(CCSTATICSYM)
409 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
410                 $(CCSTATICSYM)
411 i386_CFLAGS=    $(i386_XARCH)
412 amd64_CFLAGS=   $(amd64_XARCH)

414 sparc_ASFLAGS=  $(sparc_AS_XARCH)
415 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
416 i386_ASFLAGS=   $(i386_AS_XARCH)
417 amd64_ASFLAGS=  $(amd64_AS_XARCH)

419 #
420 sparc_COPTFLAG=         -xO3
421 sparcv9_COPTFLAG=       -xO3
422 i386_COPTFLAG=          -O
423 amd64_COPTFLAG=         -xO3

425 COPTFLAG= $($(MACH)_COPTFLAG)
426 COPTFLAG64= $($(MACH64)_COPTFLAG)

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

432 # Direct the Sun Studio compiler to use a static globalization prefix based on t
433 # name of the module rather than something unique. Otherwise, objects
434 # will not build deterministically, as subsequent compilations of identical
435 # source will yeild objects that always look different.
436 #
437 # In the same spirit, this will also remove the date from the N_OPT stab.
438 CGLOBALSTATIC= -W0,-xglobalstatic

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

444 #
445 # Default debug format for Sun Studio 11 is dwarf, so force it to
446 # generate stabs.
447 #
448 DEBUGFORMAT=    -xdebugformat=stabs

450 #
451 # Flags used to build in debug mode for ctf generation.  Bugs in the Devpro
452 # compilers currently prevent us from building with cc-emitted DWARF.
453 #
454 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
455 CTF_FLAGS_i386  = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
```



```
457 CTF_FLAGS_sparcv9       = $(CTF_FLAGS_sparc)
458 CTF_FLAGS_amd64         = $(CTF_FLAGS_i386)

460 # Sun Studio produces broken userland code when saving arguments.
461 $(__GNUC)CTF_FLAGS_amd64 += $(SAVEARGS)

463 CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
464 CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
465 CTF_FLAGS       = $(CTF_FLAGS_32)

467 #
468 # Flags used with genoffsets
469 #
470 GOFLAGS = -_noecho \
471         $(CALLSYMS) \
472         $(CDWARFSTR)

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

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

480 #
481 # tradeoff time for space (smaller is better)
482 #
483 sparc_SPACEFLAG         = -xspace -W0,-Lt
484 sparcv9_SPACEFLAG       = -xspace -W0,-Lt
485 i386_SPACEFLAG          = -xspace
486 amd64_SPACEFLAG         =

488 SPACEFLAG               = $($(MACH)_SPACEFLAG)
489 SPACEFLAG64             = $($(MACH64)_SPACEFLAG)

491 #
492 # The Sun Studio 11 compiler has changed the behaviour of integer
493 # wrap arounds and so a flag is needed to use the legacy behaviour
494 # (without this flag panics/hangs could be exposed within the source).
495 #
496 sparc_IROPTFLAG         = -W2,-xwrap_int
497 sparcv9_IROPTFLAG       = -W2,-xwrap_int
498 i386_IROPTFLAG          =
499 amd64_IROPTFLAG         =

501 IROPTFLAG               = $($(MACH)_IROPTFLAG)
502 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)

504 sparc_XREGSFLAG         = -xregs=no%appl
505 sparcv9_XREGSFLAG       = -xregs=no%appl
506 i386_XREGSFLAG          =
507 amd64_XREGSFLAG         =

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

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

517 #
518 # These variables are intended ONLY for use by developers to safely pass extra
519 # flags to the compilers without unintentionally overriding Makefile-set
520 # flags.  They should NEVER be set to any value in a Makefile.
521 #
522 # They come last in the associated FLAGS variable such that they can
```

```
523 # explicitly override things if necessary, there are gaps in this, but it's
524 # the best we can manage.
525 #
526 CUSERFLAGS              =
527 CUSERFLAGS64            = $(CUSERFLAGS)
528 CCUSERFLAGS             =
529 CCUSERFLAGS64           = $(CCUSERFLAGS)

531 CSOURCEDEBUGFLAGS       =
532 CCSOURCEDEBUGFLAGS      =
533 $(SRCDBGBLD)CSOURCEDEBUGFLAGS   = -g -xs
534 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS  = -g -xs

536 CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
537                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
538                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
539                 $(CUSERFLAGS)
540 CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
541                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
542                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
543                 $(CUSERFLAGS64)
544 #
545 # Flags that are used to build parts of the code that are subsequently
546 # run on the build machine (also known as the NATIVE_BUILD).
547 #
548 NATIVE_CFLAGS=  $(COPTFLAG) $($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
549                 $(ILDOFF) $(CERRWARN) $(C99MODE) $($(NATIVE_MACH)_CCUNBOUND) \
550                 $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
551                 $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

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

565 #
566 # For source message catalogue
567 #
568 .SUFFIXES: $(SUFFIXES) .i .po
569 MSGROOT= $(ROOT)/catalog
570 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
571 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
572 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
573 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

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

587 #
588 # This is overwritten by local Makefile when PROG is a list.
```



```
589 #
590 POFILE= $(PROG).po

592 sparc_CCFLAGS=          -cg92 -compat=4 \
593                         -Qoption ccfe -messages=no%anachronism \
594                         $(CCERRWARN)
595 sparcv9_CCFLAGS=        $(sparcv9_XARCH) -dalign -compat=5 \
596                         -Qoption ccfe -messages=no%anachronism \
597                         -Qoption ccfe -features=no%conststrings \
598                         $(CCCREGSYM) \
599                         $(CCERRWARN)
600 i386_CCFLAGS=           -compat=4 \
601                         -Qoption ccfe -messages=no%anachronism \
602                         -Qoption ccfe -features=no%conststrings \
603                         $(CCERRWARN)
604 amd64_CCFLAGS=          $(amd64_XARCH) -compat=5 \
605                         -Qoption ccfe -messages=no%anachronism \
606                         -Qoption ccfe -features=no%conststrings \
607                         $(CCERRWARN)

609 sparc_CCOPTFLAG=        -O
610 sparcv9_CCOPTFLAG=      -O
611 i386_CCOPTFLAG=         -O
612 amd64_CCOPTFLAG=        -O

614 CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
615 CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
616 CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
617                 $(CCUSERFLAGS)
618 CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
619                 $(CCUSERFLAGS64)

621 #
622 #
623 #
624 ELFWRAP_FLAGS   =
625 ELFWRAP_FLAGS64 =       -64

627 #
628 # Various mapfiles that are used throughout the build, and delivered to
629 # /usr/lib/ld.
630 #
631 MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
632 MAPFILE.NED_sparc =
633 MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
634 MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
635 MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
636 MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
637 MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy

639 #
640 # Generated mapfiles that are compiler specific, and used throughout the
641 # build.  These mapfiles are not delivered in /usr/lib/ld.
642 #
643 MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
644 $(__GNUC64)MAPFILE.NGB_sparc= \
645                         $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
646 MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
647 $(__GNUC64)MAPFILE.NGB_sparcv9= \
648                         $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
649 MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
650 $(__GNUC64)MAPFILE.NGB_i386= \
651                         $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
652 MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
653 $(__GNUC64)MAPFILE.NGB_amd64= \
654                         $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
```

```
MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

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

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

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

#
# Define compilation macros.
#
COMPILE.c=      $(CC) $(CFLAGS) $(CPPFLAGS) -c
COMPILE64.c=    $(CC) $(CFLAGS64) $(CPPFLAGS) -c
COMPILE.cc=     $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
COMPILE64.cc=   $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
COMPILE.s=      $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
COMPILE64.s=    $(AS) $(ASFLAGS) $($(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
COMPILE.d=      $(DTRACE) -G -32
COMPILE64.d=    $(DTRACE) -G -64
COMPILE.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
COMPILE64.b=    $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

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

#
# Link time macros
#
CCNEEDED                = -lC
CCEXTNEEDED             = -lCrun -lCstd
$(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -lstdc++ -lgcc_s
$(__GNUC)CCEXTNEEDED    = $(CCNEEDED)

LINK.c=         $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
LINK64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
NORUNPATH=      -norunpath -nolib
LINK.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)
LINK64.cc=      $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)

#
# lint macros
#
# Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
# ON is built with a version of lint that has the fix for 4484186.
#
```



```
ALWAYS_LINT_DEFS =      -errtags=yes -s
ALWAYS_LINT_DEFS +=     -erroff=E_PTRDIFF_OVERFLOW
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_NARROW_CONV
ALWAYS_LINT_DEFS +=     -U__PRAGMA_REDEFINE_EXTNAME
ALWAYS_LINT_DEFS +=     $(C99LMODE)
ALWAYS_LINT_DEFS +=     -errsecurity=$(SECLEVEL)
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_CREAT_WITHOUT_EXCL
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_FORBIDDEN_WARN_CREAT
# XX64 -- really only needed for amd64 lint
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_CONST_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_TO_PTR_FROM_INT
ALWAYS_LINT_DEFS +=     -erroff=E_COMP_INT_WITH_LARGE_INT
ALWAYS_LINT_DEFS +=     -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
ALWAYS_LINT_DEFS +=     -erroff=E_PASS_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_PTR_CONV_LOSES_BITS

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

SECLEVEL=       core
LINT.c=         $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT64.c=       $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT.s=         $(LINT.c)

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

# Define native compilation macros
#

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

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

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

sparcv9_CC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
```

```
 787 $(__GNUC64)sparcv9_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
 788 sparcv9_CCC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
 789 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
 790 sparcv9_CPP=            /usr/ccs/lib/cpp
 791 sparcv9_AS=             /usr/ccs/bin/as -xregsym=no
 792 sparcv9_LD=             /usr/ccs/bin/ld
 793 sparcv9_LINT=           $(SPRO_ROOT)/sunstudio12.1/bin/lint

 795 i386_CC=                $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
 796 $(__GNUC)i386_CC=       $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
 797 i386_CCC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
 798 $(__GNUC)i386_CCC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
 799 i386_CPP=               /usr/ccs/lib/cpp
 800 i386_AS=                /usr/ccs/bin/as
 801 $(__GNUC)i386_AS=       $(ONBLD_TOOLS)/bin/$(MACH)/aw
 802 i386_LD=                /usr/ccs/bin/ld
 803 i386_LINT=              $(SPRO_ROOT)/sunstudio12.1/bin/lint

 805 amd64_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
 806 $(__GNUC64)amd64_CC=    $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
 807 amd64_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
 808 $(__GNUC64)amd64_CCC=   $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
 809 amd64_CPP=              /usr/ccs/lib/cpp
 810 amd64_AS=               $(ONBLD_TOOLS)/bin/$(MACH)/aw
 811 amd64_LD=               /usr/ccs/bin/ld
 812 amd64_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

 814 NATIVECC=               $($(NATIVE_MACH)_CC)
 815 NATIVECCC=              $($(NATIVE_MACH)_CCC)
 816 NATIVECPP=              $($(NATIVE_MACH)_CPP)
 817 NATIVEAS=               $($(NATIVE_MACH)_AS)
 818 NATIVELD=               $($(NATIVE_MACH)_LD)
 819 NATIVELINT=            $($(NATIVE_MACH)_LINT)

 821 #
 822 # Makefile.master.64 overrides these settings
 823 #
 824 CC=                     $(NATIVECC)
 825 CCC=                    $(NATIVECCC)
 826 CPP=                    $(NATIVECPP)
 827 AS=                     $(NATIVEAS)
 828 LD=                     $(NATIVELD)
 829 LINT=                   $(NATIVELINT)

 831 # The real compilers used for this build
 832 CW_CC_CMD=              $(CC) -_compiler
 833 CW_CCC_CMD=             $(CCC) -_compiler
 834 REAL_CC=                $(CW_CC_CMD:sh)
 835 REAL_CCC=               $(CW_CCC_CMD:sh)

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

 840 BDIRECT=        -Bdirect
 841 BDYNAMIC=       -Bdynamic
 842 BLOCAL=         -Blocal
 843 BNODIRECT=      -Bnodirect
 844 BREDUCE=        -Breduce
 845 BSTATIC=        -Bstatic

 847 ZDEFS=          -zdefs
 848 ZDIRECT=        -zdirect
 849 ZIGNORE=        -zignore
 850 ZINITFIRST=     -zinitfirst
 851 ZINTERPOSE=     -zinterpose
 852 ZLAZYLOAD=      -zlazyload
```



```
 853 ZLOADFLTR=      -zloadfltr
 854 ZMULDEFS=       -zmuldefs
 855 ZNODEFAULTLIB=  -znodefaultlib
 856 ZNODEFS=        -znodefs
 857 ZNODELETE=      -znodelete
 858 ZNODLOPEN=      -znodlopen
 859 ZNODUMP=        -znodump
 860 ZNOLAZYLOAD=    -znolazyload
 861 ZNOLDYNSYM=     -znoldynsym
 862 ZNORELOC=       -znoreloc
 863 ZNOVERSION=     -znoversion
 864 ZRECORD=        -zrecord
 865 ZREDLOCSYM=     -zredlocsym
 866 ZTEXT=          -ztext
 867 ZVERBOSE=       -zverbose

 869 GSHARED=        -G
 870 CCMT=           -mt

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

 875 sparc_C_PICFLAGS =      -K pic
 876 sparcv9_C_PICFLAGS =    -K pic
 877 i386_C_PICFLAGS =       -K pic
 878 amd64_C_PICFLAGS =      -K pic
 879 C_PICFLAGS =            $($(MACH)_C_PICFLAGS)
 880 C_PICFLAGS64 =          $($(MACH64)_C_PICFLAGS)

 882 sparc_C_BIGPICFLAGS =   -K PIC
 883 sparcv9_C_BIGPICFLAGS = -K PIC
 884 i386_C_BIGPICFLAGS =    -K PIC
 885 amd64_C_BIGPICFLAGS =   -K PIC
 886 C_BIGPICFLAGS =         $($(MACH)_C_BIGPICFLAGS)
 887 C_BIGPICFLAGS64 =       $($(MACH64)_C_BIGPICFLAGS)

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

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

 900 #
 901 # Default label for CTF sections
 902 #
 903 CTFCVTFLAGS=           -i -L VERSION

 905 #
 906 # Override to pass module-specific flags to ctfmerge.  Currently used only by
 907 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
 908 # stripping.
 909 #
 910 CTFMRGFLAGS=

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

 914 ELFSIGN_O=      $(TRUE)
 915 ELFSIGN_CRYPTO= $(ELFSIGN_O)
 916 ELFSIGN_OBJECT= $(ELFSIGN_O)

 918 # Rules (normally from make.rules) and macros which are used for post
```

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

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

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

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

#
# chk4ubin is a tool that inspects a module for a symbol table
```



```
# ELF section size which can trigger an OBP bug on older platforms.
# This problem affects only specific sun4u bootable modules.
#
CHK4UBIN=       $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
CHK4UBINFLAGS=
CHK4UBINARY=    $(CHK4UBIN) $(CHK4UBINFLAGS) $@

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

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

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

```
1051         $(LEX.l) $< > $*.c
1052         $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1053         $(POST_PROCESS)
1054         $(RM) $*.c
1055 .l.o:
1056         $(RM) $*.c
1057         $(LEX.l) $< > $*.c
1058         $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1059         $(POST_PROCESS_O)
1060         $(RM) $*.c

1062 .bin.o:
1063         $(COMPILE.b) -o $@ $<
1064         $(POST_PROCESS_O)

1066 .java.class:
1067         $(COMPILE.java) $<

1069 # Bourne and Korn shell script message catalog build rules.
1070 # We extract all gettext strings with sed(1) (being careful to permit
1071 # multiple gettext strings on the same line), weed out the dups, and
1072 # build the catalogue with awk(1).

1074 .sh.po .ksh.po:
1075         $(SED) -n -e ":a"                                       \
1076                   -e "h"                                        \
1077                   -e "s/.*gettext *\(\"[^\"]*\"\).*/\1/p"       \
1078                   -e "x"                                        \
1079                   -e "s/\(.*\)gettext *\"[^\"]*\"\(.*\)/\1\2/"  \
1080                   -e "t a"                                      \
1081                $< | sort -u | awk '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1083 #
1084 # Python and Perl executable and message catalog build rules.
1085 #
1086 .SUFFIXES: .pl .pm .py .pyc

1088 .pl:
1089         $(RM) $@;
1090         $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1091         $(CHMOD) +x $@

1093 .py:
1094         $(RM) $@; $(CAT) $< > $@; $(CHMOD) +x $@

1096 .py.pyc:
1097         $(RM) $@
1098         $(PYTHON) -mpy_compile $<
1099         @[ $(<)c = $@ ] || $(MV) $(<)c $@

1101 .py.po:
1102         $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;

1104 .pl.po .pm.po:
1105         $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1106         $(RM)   $@ ;
1107         $(SED) "/^domain/d" < $(<F).po > $@ ;
1108         $(RM) $(<F).po

1110 #
1111 # When using xgettext, we want messages to go to the default domain,
1112 # rather than the specified one.  This special version of the
1113 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1114 # causing xgettext to put all messages into the default domain.
1115 #
1116 CPPFORPO=$(COMPILE.cpp:\"$(TEXT_DOMAIN)\"=TEXT_DOMAIN)
```



```
1118 .c.i:
1119         $(CPPFORPO) $< > $@

1121 .h.i:
1122         $(CPPFORPO) $< > $@

1124 .y.i:
1125         $(YACC) -d $<
1126         $(CPPFORPO) y.tab.c  > $@
1127         $(RM) y.tab.c

1129 .l.i:
1130         $(LEX) $<
1131         $(CPPFORPO) lex.yy.c  > $@
1132         $(RM) lex.yy.c

1134 .c.po:
1135         $(CPPFORPO) $< > $<.i
1136         $(BUILD.po)

1138 .cc.po:
1139         $(CPPFORPO) $< > $<.i
1140         $(BUILD.po)

1142 .y.po:
1143         $(YACC) -d $<
1144         $(CPPFORPO) y.tab.c  > $<.i
1145         $(BUILD.po)
1146         $(RM) y.tab.c

1148 .l.po:
1149         $(LEX) $<
1150         $(CPPFORPO) lex.yy.c  > $<.i
1151         $(BUILD.po)
1152         $(RM) lex.yy.c

1154 #
1155 # Rules to perform stylistic checks
1156 #
1157 .SUFFIXES: .x .xml .check .xmlchk

1159 .h.check:
1160         $(DOT_H_CHECK)

1162 .x.check:
1163         $(DOT_X_CHECK)

1165 .xml.xmlchk:
1166         $(MANIFEST_CHECK)

1168 #
1169 # Include rules to render automated sccs get rules "safe".
1170 #
1171 include $(SRC)/Makefile.noget
```

```
**********************************************************
    1658 Wed Sep  2 16:30:34 2015
new/usr/src/cmd/mdb/sparc/v7/libfksmbsrv/Makefile
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License, Version 1.0 only
# (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 2005 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
# Copyright 2015 Gary Mills
#

MODULE = libfksmbsrv.so
MDBTGT = proc

MODSRCS = smbsrv.c list.c

include ../../../../Makefile.cmd
include ../../Makefile.sparcv7
include ../../../Makefile.module

MODSRCS_DIR = ../../../common/modules/smbsrv
GENUNIX_DIR = ../../../common/modules/genunix

# Note: need our sys includes _before_ ENVCPPFLAGS, proto etc.
CPPFLAGS.first += -I../../../../../lib/smbsrv/libfksmbsrv/common
CPPFLAGS.first += -I../../../../../lib/libfakekernel/common

CPPFLAGS += -I../../../../../uts/common

C99MODE=        -xc99=%all
C99LMODE=       -Xc99=%all

dmod/%.o: $(GENUNIX_DIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.ln: $(GENUNIX_DIR)/%.c
        $(LINT.c) -c $<
```

```
**********************************************************
    1694 Wed Sep  2 16:30:34 2015
new/usr/src/cmd/mdb/sparc/v9/libfksmbsrv/Makefile
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License, Version 1.0 only
# (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 2005 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
# Copyright 2015 Gary Mills
#

MODULE = libfksmbsrv.so
MDBTGT = proc

MODSRCS = smbsrv.c list.c

include ../../../../Makefile.cmd
include ../../../../Makefile.cmd.64
include ../../../Makefile.sparcv9
include ../../../Makefile.module

MODSRCS_DIR = ../../../common/modules/smbsrv
GENUNIX_DIR = ../../../common/modules/genunix

# Note: need our sys includes _before_ ENVCPPFLAGS, proto etc.
CPPFLAGS.first += -I../../../../../lib/smbsrv/libfksmbsrv/common
CPPFLAGS.first += -I../../../../../lib/libfakekernel/common

CPPFLAGS += -I../../../../../uts/common

C99MODE=        -xc99=%all
C99LMODE=       -Xc99=%all

dmod/%.o: $(GENUNIX_DIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.ln: $(GENUNIX_DIR)/%.c
        $(LINT.c) -c $<
```

```
**********************************************************
    2937 Wed Sep  2 16:30:34 2015
new/usr/src/cmd/sgs/lex/Makefile.com
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
```

```
#
# 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 2015 Gary Mills
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

PROG=           lex

MACHOBJS=       main.o sub1.o sub2.o sub3.o header.o parser.o
POBJECTS=       $(MACHOBJS)
POBJS=          $(POBJECTS:%=objs/%)

LIBRARY=        libl.a
VERS=           .1

LIBOBJS=        allprint.o libmain.o reject.o yyless.o yywrap.o
LIBOBJS_W=      allprint_w.o reject_w.o yyless_w.o
LIBOBJS_E=      reject_e.o yyless_e.o
OBJECTS=        $(LIBOBJS) $(LIBOBJS_W) $(LIBOBJS_E)

FORMS=          nceucform ncform nrform

include         ../../../../lib/Makefile.lib

SRCDIR =        ../common

C99MODE=        $(C99_ENABLE)

CERRWARN +=     -_gcc=-Wno-unused-label
CERRWARN +=     -_gcc=-Wno-uninitialized
CERRWARN +=     -_gcc=-Wno-parentheses

# Override default source file derivation rule (in Makefile.lib)
# from objects
#
MACHSRCS=       $(MACHOBJS:%.o=../common/%.c)
LIBSRCS =       $(LIBOBJS:%.o=../common/%.c)
SRCS=           $(MACHSRCS) $(LIBSRCS)

LIBS =          $(DYNLIB) $(LINTLIB)
```



```
LINTSRCS=       ../common/llib-l$(LIBNAME)

INCLIST=        $(INCLIST_$(MACH)) -I../../include -I../../include/$(MACH)
DEFLIST=        -DELF

# It is not very clean to base the conditional definitions as below, but
# this will have to do for now.
#
#$(LIBOBJS_W):= DEFLIST = -DEUC -DJLSLEX  -DWOPTION -D$*=$*_w
objs/%_w.o:=    DEFLIST = -DEUC -DJLSLEX  -DWOPTION -D$*=$*_w
pics/%_w.o:=    DEFLIST = -DEUC -DJLSLEX  -DWOPTION -D$*=$*_w

#$(LIBOBJS_E):= DEFLIST = -DEUC -DJLSLEX  -DEOPTION -D$*=$*_e
objs/%_e.o:=    DEFLIST = -DEUC -DJLSLEX  -DEOPTION -D$*=$*_e
pics/%_e.o:=    DEFLIST = -DEUC -DJLSLEX  -DEOPTION -D$*=$*_e

CPPFLAGS=       $(INCLIST) $(DEFLIST) $(CPPFLAGS.master)
BUILD.AR=       $(AR) $(ARFLAGS) $@ '$(LORDER) $(OBJS) | $(TSORT)'
LINTFLAGS=      -nuaxm
LINTFLAGS64=    -nuaxm -m64
LINTFLAGS=      -amux
LINTPOUT=       lint.out

$(LINTLIB):=    LINTFLAGS = -nvx
$(ROOTPROG):=   FILEMODE = 0555

ROOTFORMS=      $(FORMS:%=$(ROOTSHLIBCCS)/%)

ROOTLINTDIR=    $(ROOTLIBDIR)
ROOTLINT=       $(LINTSRCS:../common/%=$(ROOTLINTDIR)/%)

DYNLINKLIBDIR=  $(ROOTLIBDIR)
DYNLINKLIB=     $(LIBLINKS:%=$(DYNLINKLIBDIR)/%)

# Need to make sure lib-make's are warning free
$(DYNLIB) :=    CFLAGS += $(CCVERBOSE)
$(DYNLIB) :=    CFLAGS64 += $(CCVERBOSE)

$(DYNLIB) :=    LDLIBS += -lc

CLEANFILES +=   ../common/parser.c $(LINTPOUT)
CLOBBERFILES += $(LIBS) $(LIBRARY)
```

```
**********************************************************
    3295 Wed Sep  2 16:30:34 2015
new/usr/src/cmd/sgs/libelf/Makefile.com
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
#
# 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 2015 Gary Mills
# Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
#

LIBRARY=        libelf.a
VERS=           .1
M4=             m4

MACHOBJS=
COMOBJS=        ar.o            begin.o         cntl.o          cook.o \
                data.o          end.o           fill.o          flag.o \
                getarhdr.o      getarsym.o      getbase.o       getdata.o \
                getehdr.o       getident.o      getphdr.o       getscn.o \
                getshdr.o \
                getphnum.o      getshnum.o      getshstrndx.o \
                hash.o          input.o         kind.o \
                ndxscn.o        newdata.o       newehdr.o       newphdr.o \
                newscn.o        next.o          nextscn.o       output.o \
                rand.o          rawdata.o       rawfile.o       rawput.o \
                strptr.o        update.o        error.o         gelf.o \
                clscook.o       checksum.o
CLASSOBJS=      clscook64.o     newehdr64.o     newphdr64.o     update64.o \
                checksum64.o
BLTOBJS=        msg.o           xlate.o         xlate64.o
MISCOBJS=       String.o        args.o          demangle.o      nlist.o \
                nplist.o
MISCOBJS64=     nlist.o

OBJECTS=        $(BLTOBJS)  $(MACHOBJS)  $(COMOBJS)  $(CLASSOBJS) $(MISCOBJS)

include $(SRC)/lib/Makefile.lib

# Use the value of M4 set in Makefile.master via Makefile.lib

DEMOFILES=      Makefile        00README        acom.c          dcom.c \
                pcom.c          tpcom.c         dispsyms.c
DEMOFILESRCDIR= ../demo
ROOTDEMODIRBASE=$(ROOT)/usr/demo/ELF
ROOTDEMODIRS=   $(ROOTDEMODIRBASE)
```



```
include $(SRC)/cmd/sgs/Makefile.com

WARLOCKFILES=   $(OBJECTS:%.o=wlocks/%.ll)

MAPFILES =      ../common/mapfile-vers

DYNFLAGS +=     $(VERSREF)
LDLIBS +=       $(CONVLIBDIR) $(CONV_LIB) -lc

LINTFLAGS +=    -u -erroff=E_BAD_PTR_CAST_ALIGN
LINTFLAGS64 +=  -u -erroff=E_CAST_INT_TO_SMALL_INT

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

BUILD.AR=       $(RM) $@ ; \
                $(AR) q $@ `$(LORDER) $(OBJECTS:%=$(DIR)/%)| $(TSORT)`
                $(POST_PROCESS_A)


BLTDEFS=        msg.h
BLTDATA=        msg.c
BLTMESG=        $(SGSMSGDIR)/libelf

BLTFILES=       $(BLTDEFS) $(BLTDATA) $(BLTMESG)

SGSMSGCOM=      ../common/libelf.msg
SGSMSG32=       ../common/libelf.32.msg
SGSMSGTARG=     $(SGSMSGCOM)
SGSMSGALL=      $(SGSMSGCOM) $(SGSMSG32)

SGSMSGFLAGS1=   $(SGSMSGFLAGS) -m $(BLTMESG)
SGSMSGFLAGS2=   $(SGSMSGFLAGS) -h $(BLTDEFS) -d $(BLTDATA) -n libelf_msg

BLTSRCS=        $(BLTOBJS:%.o=%.c)
LIBSRCS=        $(COMOBJS:%.o=../common/%.c)  $(MISCOBJS:%.o=../misc/%.c) \
                $(MACHOBJS:%.o=%.c)  $(BLTSRCS)
SRCS=           ../common/llib-lelf
LINTSRCS=       $(LIBSRCS) ../common/lintsup.c

ROOTFS_DYNLIB=          $(DYNLIB:%=$(ROOTFS_LIBDIR)/%)
ROOTFS_LINTLIB=         $(LINTLIB:%=$(ROOTFS_LIBDIR)/%)

ROOTFS_DYNLIB64=        $(DYNLIB:%=$(ROOTFS_LIBDIR64)/%)
ROOTFS_LINTLIB64=       $(LINTLIB:%=$(ROOTFS_LIBDIR64)/%)

$(ROOTFS_DYNLIB) :=     FILEMODE= 755
$(ROOTFS_DYNLIB64) :=   FILEMODE= 755

LIBS =          $(DYNLIB) $(LINTLIB)

CLEANFILES +=   $(LINTOUTS) $(BLTSRCS) $(BLTFILES) $(WARLOCKFILES)

.PARALLEL:      $(LIBS)
```

```
**********************************************************
    2340 Wed Sep  2 16:30:34 2015
new/usr/src/cmd/sgs/yacc/Makefile.com
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
#
# 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 2015 Gary Mills
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

PROG=           yacc

COMOBJS=        y1.o y2.o y3.o y4.o
POBJECTS=       $(COMOBJS)
POBJS=          $(POBJECTS:%=objs/%)

OBJECTS=        libmai.o libzer.o

LIBRARY=        liby.a
VERS=           .1
YACCPAR=        yaccpar

include ../../../../lib/Makefile.lib

SRCDIR =        ../common

# Override default source file derivation rule (in Makefile.lib)
# from objects
#
COMSRCS=        $(COMOBJS:%.o=../common/%.c)
LIBSRCS=        $(OBJECTS:%.o=../common/%.c)
SRCS=           $(COMSRCS) $(LIBSRCS)

LIBS =          $(DYNLIB) $(LINTLIB)

# Tune ZDEFS to ignore undefined symbols for building the yacc shared library
# since these symbols (mainly yyparse) are to be resolved elsewhere.
#
$(DYNLIB):= ZDEFS = $(ZNODEFS)
$(DYNLIBCCC):= ZDEFS = $(ZNODEFS)
LINTSRCS=       ../common/llib-l$(LIBNAME)

INCLIST=        -I../../include -I../../include/$(MACH)
CPPFLAGS=       $(INCLIST) $(DEFLIST) $(CPPFLAGS.master)
LDLIBS=         $(LDLIBS.cmd)
```



```
BUILD.AR=       $(AR) $(ARFLAGS) $@ '$(LORDER) $(OBJS) | $(TSORT)'
LINTFLAGS=      -nuaxm
LINTFLAGS64=    -nuaxm -m64
LINTFLAGS=      -amux
LINTPOUT=       lint.out

C99MODE= $(C99_ENABLE)
CFLAGS += $(CCVERBOSE)
CFLAGS64 += $(CCVERBOSE)
CERRWARN += -_gcc=-Wno-parentheses
CERRWARN += -_gcc=-Wno-uninitialized

$(LINTLIB):=    LINTFLAGS = -nvx
$(ROOTPROG):= FILEMODE = 0555

ROOTYACCPAR=    $(YACCPAR:%=$(ROOTSHLIBCCS)/%)

ROOTLINTDIR=    $(ROOTLIBDIR)
ROOTLINT=       $(LINTSRCS:../common/%=$(ROOTLINTDIR)/%)

DYNLINKLIBDIR=  $(ROOTLIBDIR)
DYNLINKLIB=     $(LIBLINKS:%=$(DYNLINKLIBDIR)/%)

$(DYNLIB) :=    LDLIBS += -lc

CLEANFILES +=   $(LINTPOUT)
CLOBBERFILES += $(LIBS) $(LIBRARY)
```

```
**********************************************************
   14029 Wed Sep  2 16:30:34 2015
new/usr/src/lib/Makefile
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
```

```
   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 # Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright (c) 2012 by Delphix. All rights reserved.
  25 # Copyright (c) 2012, Joyent, Inc. All rights reserved.
  26 # Copyright (c) 2013 Gary Mills
  27 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  28 # Copyright (c) 2015 Gary Mills

  30 include ../Makefile.master

  32 #       Note that libcurses installs commands along with its library.
  33 #       This is a minor bug which probably should be fixed.
  34 #       Note also that a few extra libraries are kept in cmd source.
  35 #
  36 # Certain libraries are linked with, hence depend on, other libraries.
  37 #
  38 # Although we have historically used .WAIT to express dependencies, it
  39 # reduces the amount of parallelism and thus lengthens the time it
  40 # takes to build the libraries.  Thus, we now require that any new
  41 # libraries explicitly call out their dependencies.  Eventually, all
  42 # the library dependencies will be called out explicitly.  See
  43 # "Library interdependencies" near the end of this file.
  44 #
  45 # Aside from explicit dependencies (and legacy .WAITs), all libraries
  46 # are built in parallel.
  47 #
  48 .PARALLEL:

  50 SUBDIRS= \
  51         common                  .WAIT   \
  52         ../cmd/sgs/libconv              \
  53         ../cmd/sgs/libdl        .WAIT

  55 SUBDIRS += \
  56         libc                    .WAIT   \
  57         ../cmd/sgs/libelf       .WAIT   \
  58         c_synonyms                      \
  59         libmd                           \
  60         libmd5                          \
  61         librsm                          \
```



```
  62         libmp                   .WAIT   \
  63         libnsl                  \
  64         libsecdb                .WAIT   \
  65         librpcsvc                       \
  66         libsocket               .WAIT   \
  67         libsctp                 \
  68         libsip                  \
  69         libcommputil            \
  70         libresolv                       \
  71         libresolv2              .WAIT   \
  72         libw                    .WAIT   \
  73         libintl                 .WAIT   \
  74         ../cmd/sgs/librtld_db   \
  75         libaio                  \
  76         libast                  \
  77         libdll                  \
  78         libcmd                  \
  79         libshell                \
  80         libsum                  \
  81         librt                   \
  82         libadm                  \
  83         libctf                  \
  84         libdtrace               \
  85         libdtrace_jni           \
  86         libcurses               \
  87         libtermcap              \
  88         libgen                  \
  89         libgss                  \
  90         libpam                  \
  91         libuuid                 \
  92         libthread               \
  93         libpthread      .WAIT   \
  94         libslp          \
  95         libbsdmalloc    \
  96         libdoor         \
  97         libdevinfo      \
  98         libdladm        \
  99         libdlpi         \
 100         libeti          \
 101         libcrypt        \
 102         libdns_sd       \
 103         libefi          \
 104         libfstyp        \
 105         libwanboot      \
 106         libwanbootutil  \
 107         libcryptoutil   \
 108         libinetutil     \
 109         libipadm        \
 110         libipd          \
 111         libipmp         \
 112         libiscsit       \
 113         libkmf          \
 114         libkstat        \
 115         libkvm          \
 116         liblm           \
 117         libmalloc       \
 118         libmapmalloc    \
 119         libmtmalloc     \
 120         libnls          \
 121         libnwam         \
 122         libsmbios       \
 123         libtecla        \
 124         libumem         \
 125         libnvpair       .WAIT   \
 126         libexacct       \
 127         libsasl         \
```

```
128         libldap5        \
129         libsldap        .WAIT   \
130         libbsm          \
131         libsys          \
132         libsysevent     \
133         libnisdb        \
134         libpool         \
135         libpp           \
136         libproc         \
137         libproject      \
138         libsendfile     \
139         nametoaddr      \
140         ncad_addr       \
141         hbaapi          \
142         smhba           \
143         sun_fc          \
144         sun_sas         \
145         gss_mechs/mech_krb5     .WAIT   \
146         libkrb5 .WAIT   \
147         krb5    .WAIT   \
148         libsmbfs        \
149         libfcoe         \
150         libsrpt         \
151         libstmf         \
152         libstmfproxy    \
153         libnsctl        \
154         libunistat      \
155         libdscfg        \
156         librdc          \
157         libinstzones    \
158         libpkg          \
159         libpcidb        \
160         libm1           \
161         libm            \
162         libmvec


165 SUBDIRS += \
166         passwdutil      \
167         pam_modules     \
168         crypt_modules   \
169         libadt_jni      \
170         abi             \
171         auditd_plugins  \
172         libvolmgt       \
173         libdevice       \
174         libdevid        \
175         libc_db         \
176         libndmp         \
177         libsec          \
178         libtnfprobe     \
179         libtnf          \
180         libtnfctl       \
181         libdhcpagent    \
182         libdhcputil     \
183         libxnet         \
184         libipsecutil    \
185         nsswitch        \
186         print           \
187         libuutil        \
188         libscf          \
189         libinetsvc      \
190         librestart      \
191         libsched        \
192         libelfsign      \
193         pkcs11          .WAIT   \
```



```
194         libpctx         .WAIT   \
195         libcpc          \
196         getloginx       \
197         watchmalloc     \
198         extendedFILE    \
199         madv            \
200         mpss            \
201         libdisasm       \
202         libwrap         \
203         libxcurses      \
204         libxcurses2     \
205         libbrand        .WAIT   \
206         libzonecfg      \
207         libzoneinfo     \
208         libzonestat     \
209         libtsnet        \
210         libtsol         \
211         gss_mechs/mech_spnego   \
212         gss_mechs/mech_dummy    \
213         gss_mechs/mech_dh       \
214         rpcsec_gss      \
215         libraidcfg      .WAIT   \
216         librcm          .WAIT   \
217         libcfgadm       .WAIT   \
218         libpicl         .WAIT   \
219         libpicltree     .WAIT \
220         raidcfg_plugins \
221         cfgadm_plugins  \
222         libmail         \
223         lvm             \
224         libsmedia       \
225         libipp          \
226         libdiskmgt      \
227         liblgrp         \
228         libfsmgt        \
229         fm              \
230         libavl          \
231         libcmdutils     \
232         libcontract     \
233         ../cmd/sendmail/libmilter       \
234         sasl_plugins    \
235         udapl           \
236         libzpool        \
237         libzfs_core     \
238         libzfs          \
239         libbe           \
240         pylibbe         \
241         libzfs_jni      \
242         pyzfs           \
243         pysolaris       \
244         libmapid        \
245         brand           \
246         policykit       \
247         hal             \
248         libshare        \
249         libsqlite       \
250         libidmap        \
251         libadutils      \
252         libipmi         \
253         libexacct/demo  \
254         libvrrpadm      \
255         libvscan        \
256         libgrubmgmt     \
257         libfakekernel   \
258         smbsrv          \
259         libilb          \
```

```
260         scsi            \
261         libima          \
262         libsun_ima      \
263         mpapi           \
264         librstp         \
265         libreparse      \
266         libhotplug      \
267         libfruutils     .WAIT   \
268         libfru          \
269         $($(MACH)_SUBDIRS)

271 i386_SUBDIRS=           \
272         libfdisk        \
273         libsaveargs

275 sparc_SUBDIRS= .WAIT    \
276         efcode          \
277         libds           \
278         libdscp         \
279         libprtdiag      .WAIT   \
280         libprtdiag_psr  \
281         libpri          \
282         librsc          \
283         storage         \
284         libpcp          \
285         libtsalarm      \
286         libv12n

288 FM_sparc_DEPLIBS= libpri

290 fm:                     \
291         libexacct       \
292         libipmi         \
293         libzfs          \
294         scsi            \
295         $(FM_$(MACH)_DEPLIBS)

297 #
298 # Create a special version of $(SUBDIRS) with no .WAIT's, for use with the
299 # clean and clobber targets (for more information, see those targets, below).
300 #
301 NOWAIT_SUBDIRS= $(SUBDIRS:.WAIT=)

303 DCSUBDIRS =             \
304         lvm

306 MSGSUBDIRS=             \
307         abi             \
308         auditd_plugins  \
309         brand           \
310         cfgadm_plugins  \
311         gss_mechs/mech_dh       \
312         gss_mechs/mech_krb5     \
313         krb5            \
314         libast          \
315         libbsm          \
316         libc            \
317         libcfgadm       \
318         libcmd          \
319         libcontract     \
320         libcurses       \
321         libdhcputil     \
322         libipsecutil    \
323         libdiskmgt      \
324         libdladm        \
325         libdll          \
```



```
326         libgrubmgmt     \
327         libgss          \
328         libidmap        \
329         libipmp         \
330         libilb          \
331         libinetutil     \
332         libinstzones    \
333         libipadm        \
334         libnsl          \
335         libnwam         \
336         libpam          \
337         libpicl         \
338         libpool         \
339         libpkg          \
340         libpp           \
341         libscf          \
342         libsasl         \
343         libldap5        \
344         libsecdb        \
345         libshare        \
346         libshell        \
347         libsldap        \
348         libslp          \
349         libsmbfs        \
350         libsmedia       \
351         libsum          \
352         libtsol         \
353         libuutil        \
354         libvrrpadm      \
355         libvscan        \
356         libwanboot      \
357         libwanbootutil  \
358         libzfs          \
359         libzonecfg      \
360         lvm             \
361         madv            \
362         mpss            \
363         pam_modules     \
364         pyzfs           \
365         pysolaris       \
366         rpcsec_gss      \
367         libreparse
368 MSGSUBDIRS += \
369         $($(MACH)_MSGSUBDIRS)

371 sparc_MSGSUBDIRS=       \
372         libprtdiag      \
373         libprtdiag_psr

375 i386_MSGSUBDIRS= libfdisk

377 HDRSUBDIRS=             \
378         auditd_plugins  \
379         libast          \
380         libbrand        \
381         libbsm          \
382         libc            \
383         libcmd          \
384         libcmdutils     \
385         libcommputil    \
386         libcontract     \
387         libcpc          \
388         libctf          \
389         libcurses       \
390         libtermcap      \
391         libcryptoutil   \
```

```
392         libdevice       \
393         libdevid        \
394         libdevinfo      \
395         libdiskmgt      \
396         libdladm        \
397         libdll          \
398         libdlpi         \
399         libdhcpagent    \
400         libdhcputil     \
401         libdisasm       \
402         libdns_sd       \
403         libdscfg        \
404         libdtrace       \
405         libdtrace_jni   \
406         libelfsign      \
407         libeti          \
408         libfakekernel   \
409         libfru          \
410         libfstyp        \
411         libgen          \
412         libipadm        \
413         libipd          \
414         libipsecutil    \
415         libinetsvc      \
416         libinetutil     \
417         libinstzones    \
418         libipmi         \
419         libipmp         \
420         libipp          \
421         libiscsit       \
422         libkstat        \
423         libkvm          \
424         libmail         \
425         libmd           \
426         libmtmalloc     \
427         libndmp         \
428         libnvpair       \
429         libnsctl        \
430         libnsl          \
431         libnwam         \
432         libpam          \
433         libpcidb        \
434         libpctx         \
435         libpicl         \
436         libpicltree     \
437         libpool         \
438         libpp           \
439         libproc         \
440         libraidcfg      \
441         librcm          \
442         librdc          \
443         libscf          \
444         libsip          \
445         libsmbios       \
446         librestart      \
447         librpcsvc       \
448         librsm          \
449         librstp         \
450         libsasl         \
451         libsec          \
452         libshell        \
453         libslp          \
454         libsmedia       \
455         libsocket       \
456         libsqlite       \
457         libfcoe         \
```



```
458         libsrpt         \
459         libstmf         \
460         libstmfproxy    \
461         libsum          \
462         libsysevent     \
463         libtecla        \
464         libtnf          \
465         libtnfctl       \
466         libtnfprobe     \
467         libtsnet        \
468         libtsol         \
469         libvrrpadm      \
470         libvolmgt       \
471         libumem         \
472         libunistat      \
473         libuutil        \
474         libwanboot      \
475         libwanbootutil  \
476         libwrap         \
477         libxcurses2     \
478         libzfs          \
479         libzfs_core     \
480         libzfs_jni      \
481         libzoneinfo     \
482         libzonestat     \
483         hal             \
484         policykit       \
485         lvm             \
486         pkcs11          \
487         passwdutil      \
488         ../cmd/sendmail/libmilter       \
489         fm              \
490         udapl           \
491         libmapid        \
492         libkrb5         \
493         libsmbfs        \
494         libshare        \
495         libidmap        \
496         libvscan        \
497         libgrubmgmt     \
498         smbsrv          \
499         libilb          \
500         scsi            \
501         hbaapi          \
502         smhba           \
503         libima          \
504         libsun_ima      \
505         mpapi           \
506         libreparse      \
507         $($(MACH)_HDRSUBDIRS)

509 i386_HDRSUBDIRS=       \
510         libfdisk        \
511         libsaveargs

513 sparc_HDRSUBDIRS=       \
514         libds           \
515         libdscp         \
516         libpri          \
517         libv12n         \
518         storage

520 all :=          TARGET= all
521 check :=        TARGET= check
522 clean :=        TARGET= clean
523 clobber :=      TARGET= clobber
```

```
524 install :=      TARGET= install
525 install_h :=    TARGET= install_h
526 lint :=         TARGET= lint
527 _dc :=          TARGET= _dc
528 _msg :=         TARGET= _msg

530 .KEEP_STATE:

532 #
533 # For the all and install targets, we clearly must respect library
534 # dependencies so that the libraries link correctly.  However, for
535 # the remaining targets (check, clean, clobber, install_h, lint, _dc
536 # and _msg), libraries do not have any dependencies on one another
537 # and thus respecting dependencies just slows down the build.
538 # As such, for these rules, we use pattern replacement to explicitly
539 # avoid triggering the dependency information.  Note that for clean,
540 # clobber and lint, we must use $(NOWAIT_SUBDIRS) rather than
541 # $(SUBDIRS), to prevent '.WAIT' from expanding to '.WAIT-nodepend'.
542 #

544 all:                    $(SUBDIRS)

546 install:                $(SUBDIRS) .WAIT install_extra

548 # extra libraries kept in other source areas
549 install_extra:
550         @cd ../cmd/sgs; pwd; $(MAKE) install_lib
551         @pwd

553 clean clobber lint:     $(NOWAIT_SUBDIRS:%=%-nodepend)

555 install_h check:        $(HDRSUBDIRS:%=%-nodepend)

557 _msg:                   $(MSGSUBDIRS:%=%-nodepend) .WAIT _dc

559 _dc:                    $(DCSUBDIRS:%=%-nodepend)

561 #
562 # Library interdependencies are called out explicitly here
563 #
564 auditd_plugins: libbsm libnsl libsecdb
565 krb5:           gss_mechs/mech_krb5
566 gss_mechs/mech_krb5:    libgss libnsl libsocket libresolv pkcs11
567 gss_mechs/mech_spnego:  gss_mechs/mech_krb5
568 libadt_jni:     libbsm
569 libast:         libsocket libm
570 libadutils:     libldap5 libresolv libsocket libnsl
571 nsswitch:       libadutils libidmap
572 libbe:          libzfs
573 libbsm:         libtsol
574 libcmd:         libsum libast libsocket libnsl
575 libcmdutils:    libavl libnvpair
572 libcmdutils:    libavl
576 libcontract:    libnvpair
577 libcpc:         libdevinfo
578 libdevid:       libdevinfo
579 libdevinfo:     libnvpair libsec
580 libdhcpagent:   libsocket libdhcputil libuuid libdlpi libcontract
581 libdhcputil:   libnsl libgen libinetutil libdlpi
582 libdladm:       libdevinfo libinetutil libsocket libscf librcm libnvpair \
583                 libexacct libnsl libkstat libcurses
584 libdevice:      libnvpair
585 libdll:         libast
586 libdlpi:        libinetutil libdladm
587 libds:          libsysevent
588 libdscfg:       libnsctl libunistat libsocket libnsl
```



```
589 libdtrace:      libproc libgen libctf libmapmalloc
584 libdtrace:      libproc libgen libctf
590 libdtrace_jni:  libuutil libdtrace
591 libefi:         libuuid
592 libfcoe:        libdladm
593 libfstyp:       libnvpair
594 libelfsign:     libcryptoutil libkmf
595 libidmap:       libadutils libldap5 libavl libsldap libuutil
596 libipadm:       libnsl libinetutil libsocket libdlpi libnvpair libdhcpagent \
597                 libdladm libsecdb
598 libiscsit:      libc libnvpair libstmf libuuid libnsl
599 libkmf:         libcryptoutil pkcs11
600 libm:           libc
601 libm1:          libc libm
602 libmvec:        libc libm
603 libnsl:         libmd5
604 libmapid:       libresolv
605 libnisdb:       libnsl libldap5
606 librcm:         libnvpair
607 librdc:         libsocket libnsl libnsctl libunistat libdscfg
608 libuuid:        libdlpi
609 libinetutil:    libsocket
610 libipsecutil:   libtecla libsocket
611 libinstzones:   libzonecfg libcontract
612 libpkg:         libwanboot libscf libadm
613 libnwam:        libscf libbsm libdladm
605 libnwam:        libscf
614 libsecdb:       libnsl
615 libsasl:        libgss libsocket pkcs11 libmd
616 sasl_plugins:   pkcs11 libgss libsocket libsasl
617 libsctp:        libsocket
618 libshell:       libast libcmd libdll libsocket libsecdb libm
619 libsip:        libmd5
620 libsmbfs:       libcmdutils libsocket libnsl libkrb5 libsec libidmap
612 libsmbfs:       libcmdutils libsocket libnsl libkrb5
621 libsocket:     libnsl
622 libstmfproxy:   libstmf libsocket libnsl libpthread
623 libsum:         libast
624 libsysevent:    libsecdb libnvpair
616 libsysevent:    libsecdb
625 libldap5:       libsasl libsocket libnsl libmd
626 libsldap:       libldap5 libtsol libnsl libc libscf libresolv
627 libpool:       libnvpair libexacct
628 libpp:         libast
629 libzonecfg:     libc libsocket libnsl libuuid libnvpair libsysevent libsec \
630                 libbrand libpool libscf
631 libproc:        ../cmd/sgs/librtld_db ../cmd/sgs/libelf libctf libsaveargs
632 libproject:     libpool libproc libsecdb
633 libtermcap:     libcurses
634 libtsnet:       libnsl libtsol libsecdb
635 libwrap:        libnsl libsocket
636 libwanboot:     libnvpair libresolv libnsl libsocket libdevinfo libinetutil \
637                 libdhcputil
638 libwanbootutil: libnsl
639 pam_modules:    libproject passwdutil smbsrv libtsnet
640 libscf:         libuutil libmd libgen libsmbios libnsl libnvpair
631 pam_modules:    libproject passwdutil smbsrv
632 libscf:         libuutil libmd libgen libsmbios libnsl
641 libinetsvc:     libscf
642 librestart:     libuutil libscf libpool libproject libsecdb libnvpair
634 librestart:     libuutil libscf
643 libsaveargs:    libdisasm
644 ../cmd/sgs/libdl:       ../cmd/sgs/libconv
645 ../cmd/sgs/libelf:      ../cmd/sgs/libconv
646 passwdutil:     libsldap
647 pkcs11:         libcryptoutil libmd
```

```
648 #               Adding libuuid or libdladm results in a circular dependancy
638 pkcs11:         libcryptoutil
649 print:          libldap5
650 udapl/udapl_tavor:      udapl/libdat
651 libzfs:         libdevid libgen libnvpair libuutil \
652                 libadm libavl libefi libidmap libmd libzfs_core libm
653 libzfs_core:    libnvpair
654 libzfs_jni:     libdiskmgt libnvpair libzfs
655 libzpool:       libavl libumem libnvpair libcmdutils
656 libsec:         libavl libidmap
657 brand:          libc libsocket
658 libshare:       libscf libzfs libuuid libfsmgt libsecdb libumem libsmbfs
659 libexacct/demo: libexacct libproject libsocket libnsl
660 libtsalarm:     libpcp
661 smbsrv:         libsocket libnsl libmd libxnet libpthread librt \
662                 libshare libidmap pkcs11 libsqlite libcryptoutil \
663                 libreparse libcmdutils libfakekernel
664 libv12n:        libds libuuid
665 libvrrpadm:     libsocket libdladm libscf
666 libvscan:       libscf
667 libfru:         libfruutils
668 scsi:           libnvpair libfru
669 mpapi:          libpthread libdevinfo libsysevent libnvpair
670 sun_fc:         libdevinfo libsysevent libnvpair
671 libsun_ima:     libdevinfo libsysevent libnsl
672 sun_sas:        libdevinfo libsysevent libnvpair libkstat libdevid
673 libgrubmgmt:    libdevinfo libzfs libfstyp
674 pylibbe:        libbe libzfs
675 pyzfs:          libnvpair libzfs
676 pysolaris:      libsec libidmap
677 libreparse:     libnvpair
678 libhotplug:     libnvpair
679 cfgadm_plugins: libhotplug
680 libilb:         libsocket
681 libipmi:        libm
682 libprtdiag:     libm
683 libsqlite:      libm
684 libstmf:        libm
685 libvscan:       libm


688 $(INTEL_BUILD)libdiskmgt:libfdisk

690 #
691 # The reason this rule checks for the existence of the
692 # Makefile is that some of the directories do not exist
693 # in certain situations (e.g., exportable source builds,
694 # OpenSolaris).
695 #
696 $(SUBDIRS): FRC
697         @if [ -f $@/Makefile  ]; then \
698                 cd $@; pwd; $(MAKE) $(TARGET); \
699         else \
700                 true; \
701         fi

703 $(SUBDIRS:%=%-nodepend):
704         @if [ -f $(@:%-nodepend=%)/Makefile  ]; then \
705                 cd $(@:%-nodepend=%); pwd; $(MAKE) $(TARGET); \
706         else \
707                 true; \
708         fi

710 FRC:
```

```
**********************************************************
    2541 Wed Sep  2 16:30:35 2015
new/usr/src/lib/libcurses/screen/keyname.sh
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
```

```
   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, Version 1.0 only
   6 # (the "License").  You may not use this file except in compliance
   7 # 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 #
  22 #
  23 # Copyright 2015 Gary Mills
  24 # Copyright 1997 Sun Microsystems, Inc.  All rights reserved.
  25 # Use is subject to license terms.
  26 #

  28 #
  29 #       Copyright (c) 1988 AT&T
  30 #         All Rights Reserved
  31 #

  33 #
  34 # University Copyright- Copyright (c) 1982, 1986, 1988
  35 # The Regents of the University of California
  36 # All Rights Reserved
  37 #
  38 # University Acknowledgment- Portions of this document are derived from
  39 # software developed by the University of California, Berkeley, and its
  40 # contributors.
  41 #
  42
  42 #ident  "%Z%%M% %I%     %E% SMI"

  43 rm -f keyname.c
  44 /usr/bin/print "#include        \"curses_inc.h\"\n" > keyname.c
  45 /usr/bin/print "static  char    *keystrings[] =\n\t\t{" >> keyname.c
  45 /usr/bin/echo "#include \"curses_inc.h\"\n" > keyname.c
  46 /usr/bin/echo "static   char    *keystrings[] =\n\t\t{" >> keyname.c
  46 {
  47     grep -v 'KEY_F(' keycaps | awk '{ print $5, $4 }' | sed -e 's/,//g' -e 's/KE
  48     # These three aren't in keycaps
  49     echo '0401 BREAK\n0530 SRESET\n0531 RESET'
  50 } |  sort -n | awk '
______unchanged_portion_omitted_
```

```
**********************************************************
   12762 Wed Sep  2 16:30:35 2015
new/usr/src/lib/libcurses/screen/maketerm.ed
6117 Many small bugs prevent a clean build on SPARC
**********************************************************
______unchanged_portion_omitted_

 286 struct strs4 {
 287     charptr
 288 .
 289 1,$s/_Sentinel/Sentinel/
 290 w ./tmp/term.h.new
 291 e ./tmp/term.h.new
 292 g/[     .]_cursor_/s/_cursor_/_crsr_/
 293 g/[     .]_delete_/s/_delete_/_dlt_/
 294 g/[     .]_enter_/s/_enter_/_entr_/
 295 g/[     .]_insert_/s/_insert_/_ins_/
 296 g/[     .]_key_/s/_key_/_ky_/
 297 g/[     .]_keypad_/s/_keypad_/_kpad_/
 298 g/[     .]_label_/s/_label_/_labl_/
 299 g/[     .]_memory_/s/_memory_/_mem_/
 300 g/[     .]_parm_/s/_parm_/_prm_/
 301 g/[     .]_scroll_/s/_scroll_/_scrll_/
 302 g/^     _......[,;]                    \//s/,  /,/
 303 g/^     _..............[,;]            \//s/,  /,/
 304 g/^     _......................[,;]     \//s/,  /,/
 305 w ./tmp/term.h.new
 306 e ./tmp/term.h.new
 307 !# the following lines GO away when Vr2 compat code goes away
 308 1;/^#define     auto_left_margin/,/^#define     xon_xoff/s/CURB.*/CUR _b/
 309 1;/^#define     auto_left_margin/,/^#define     xon_xoff/w ./tmp/tmp.term.h
 310 1;/^#define     auto_left_margin/,/^#define     xon_xoff/d
 311 !/usr/bin/print '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 311 !/usr/bin/echo '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 312 .-1r !awk -f ./tmp/tmp.awk < ./tmp/tmp.term.h
 313 1;/^#define     columns/,/^#define      width_status_line/s/CURN.*/CUR _c/
 314 1;/^#define     columns/,/^#define      width_status_line/w ./tmp/tmp.term.h
 315 1;/^#define     columns/,/^#define      width_status_line/d
 316 !/usr/bin/print '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 316 !/usr/bin/echo '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 317 .-1r !awk -f ./tmp/tmp.awk < ./tmp/tmp.term.h
 318 1;/^#define     back_tab/,/^#define     lab_f8/s/CURS.*/CUR _Vr2_Astrs._s/
 319 1;/^#define     back_tab/,/^#define     lab_f8/w ./tmp/tmp.term.h
 320 1;/^#define     back_tab/,/^#define     lab_f8/d
 321 !/usr/bin/print '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 321 !/usr/bin/echo '{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 322 .-1r !awk -f ./tmp/tmp.awk < ./tmp/tmp.term.h
 323 1;/^#define     lab_f9/,/^#define       prtr_non/s/CURS.*/CUR _Vr2_Bstrs._s/
 324 1;/^#define     lab_f9/,/^#define       prtr_non/w ./tmp/tmp.term.h
 325 1;/^#define     lab_f9/,/^#define       prtr_non/d
 326 !/usr/bin/print 'BEGIN{i=100}\n{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 326 !/usr/bin/echo 'BEGIN{i=100}\n{printf "\%s\%d\\n", $0,++i}' > ./tmp/tmp.awk
 327 .-1r !awk -f ./tmp/tmp.awk < ./tmp/tmp.term.h
 328 !rm -f ./tmp/tmp.term.h ./tmp/tmp.awk
 329 w ./tmp/term.h.new
 330 e ./tmp/term.h.new
 331 !# Vr2 compat code ends here
 332 1i
 333 #ifndef _TERM_H
 334 #define _TERM_H

 336 /*
 337  * term.h - this file is automatically made from caps and maketerm.ed.
 338  * Don't make changes directly to term.h.
 339  */
```



```
 341 #ifdef  __cplusplus
 342 extern "C" {
 343 #endif

 345 .
 346 $a

 348 struct _str_struct {
 349         struct strs strs;
 350         struct strs2 strs2;
 351         struct strs3 strs3;
 352         struct strs4 strs4;
 353 };
______unchanged_portion_omitted_
```